ОКП 42 2953

КОНТРОЛЛЕРЫ ПРОГРАММИРУЕМЫЕ ГСП МикроДАТ МБ 57.0

Инструкция по эксплуатации

Часть II

Описание языка

гШ2.598.148 ИЭ1

I. ВВЕДЕНИЕ

I.I. Общие сведения

Настоящий документ содержит сведения, необходимые для правильного использования контроллеров программируемых ГСП МикроДАТ МБ 57.0.

Подробно описывается система инструкций языка символического кодирования программируемых контроллеров (ПК), форматы представления данных и правила составления программ.

Кроме описания языка символического кодирования ПК документ содержит примеры по составлению различных программ.

I.2. Программируемые контроллеры

ПК – это электронная машина, доступная для программирования инженеру-технологу, неспециалисту в области программирования, предназначенная для управления последовательными логическими процессами в условиях промышленной среды в реальном масштабе времени. Структура ПК приведена на рис. I.I.

ПК выполняет три функции:

- получение информации со входов;
- принятие решения;
- выдачу сигналов на выходы.

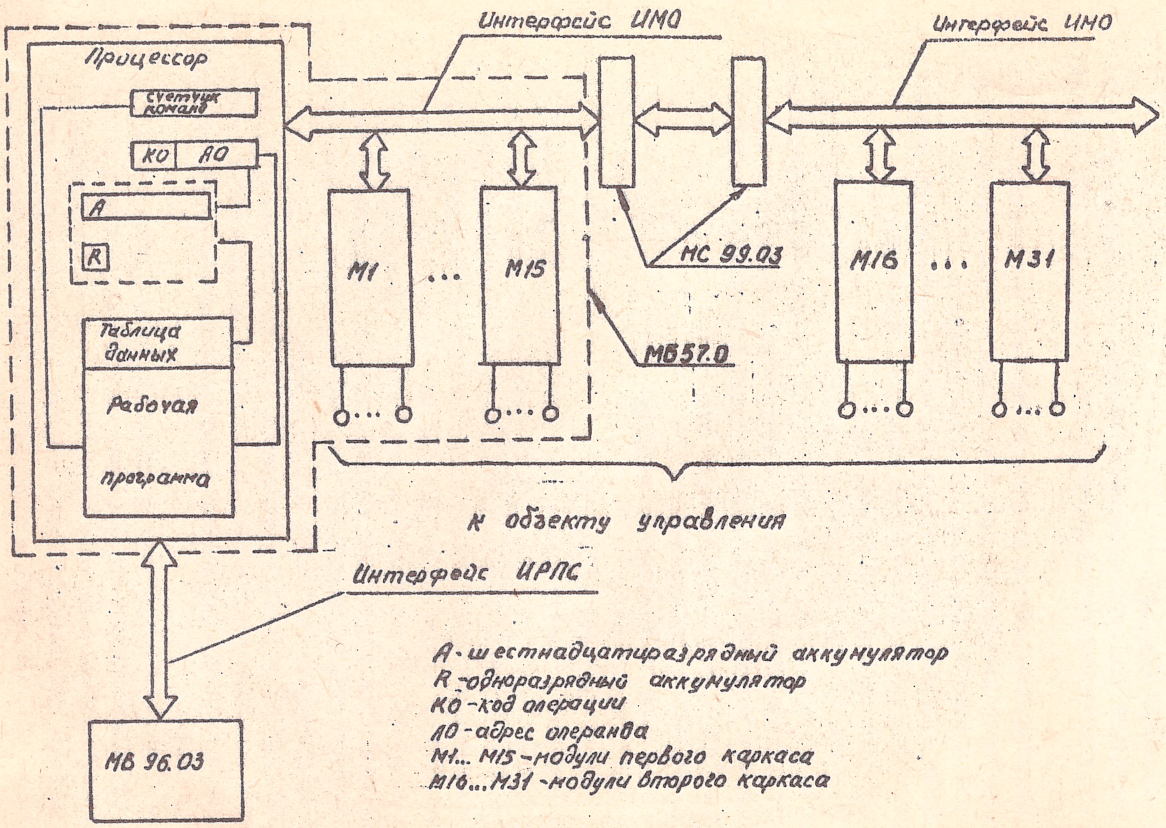
Информация представлена в виде дискретных, аналоговых, импульсных сигналов, поступающих от различного рода датчиков, о состоянии оборудования, положении механизмов и являющихся источником информации органов ручного управления.

Обработка информации и принятие решения производится центральным блоком обработки информации ПК – модулем микропроцессорным МС 59.06 или МС 59.07 (в дальнейшем – процессор).

Он постоянно опрашивает входные сигналы, поступающие от входных устройств, и производит логические операции с этими сигналами по программе, заданной пользователем.

Действия ПК состоят во включении и выключении выходов, управляющих внешними устройствами, в качестве которых могут быть применены пускатели, устройства сигнализации и индикации, логические схемы.

Автоматизация объекта управления (машины, группы машин или любого технологического оборудования вообще) заключается в обеспечении его работы с помощью устройства управления. Созданная таким образом автоматизированная система функционирует с учетом параметров состояния объекта управления. Оператор осуществляет



A - шестнадцатиразрядный аккумулятор
R - одноразрядный аккумулятор
КО - код операции
АО - адрес операнда
М1... М15 - модули первого каркаса
М16... М31 - модули второго каркаса

Рис. 1.1

Заказ 78. Тираж 1500. 25. 01. 1988г.

СОДЕРЖАНИЕ

	Лист
I. Введение	3
I.I. Общие сведения	3
I.2. Программируемые контроллеры	3
I.3. Основные логические схемы	5
2. Описание языка	7
2.I. Общие сведения	7
2.2. Структура памяти ПК	10
2.2.I. Таблица данных	10
2.2.2. Область задания маски обмена	19
2.2.3. Область задания конфигурации	19
2.2.4. Область задания текущего состояния	19
2.2.5. Область диагностики отказов и сбоев	19
2.3. Структура программы	22
2.4. Представление числа	30
2.5. Инструкции логические	30
2.6. Инструкции структурирования	38
2.7. Инструкции для работы с подпрограммами	42
2.8. Инструкции таймеров и счетчиков	43
2.9. Инструкции для работы с данными	53
2.I0. Инструкции сравнения	59
2.II. Файловые инструкции	61
2.I2. Инструкции экстренного обращения к модулям ввода-вывода	66
2.I3. Арифметические инструкции	67
2.I4. Примеры по составлению рабочих программ	69
Приложение. Список инструкций языка символического кодирования	80

лишь общий контроль за ходом процесса, наблюдает за работой машин и при необходимости берет на себя управление работой всей системы или ее части.

Алгоритм управления задается ПК в виде программы, которую легко заменить или модифицировать.

I.3. Основные логические схемы

Логической схемой называют такую схему, которая выполняет логические операции над одной или более логическими переменными. Логическая переменная - это электрический сигнал, принимающий два различных значения, известные как состояния ИСТИНА и ЛОЖЬ.

Например, логическая переменная А может быть представлена состоянием однополюсного переключателя, который может быть либо разомкнут, либо замкнут; следовательно, А принимает два значения. Значению ИСТИНА переменной А соответствует замкнутое состояние переключателя, значению ЛОЖЬ - разомкнутое (рис. I.2). Иногда истинное и ложное состояния обозначают соответственно терминами "высокий - низкий уровни", "да - нет", "включено-выключено", "логическая I - логический 0".

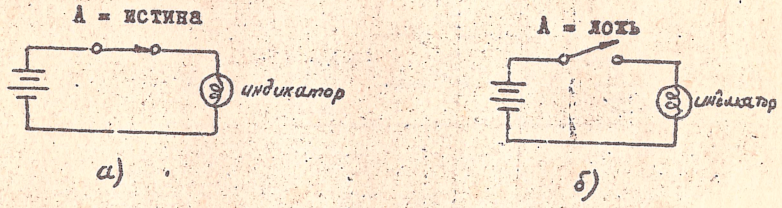


Рис. I.2

Ниже рассмотрены схемы, реализующие операции двоичной логики.

СХЕМА И

Схема И выполняет операцию логического умножения (конъюнкции) двух или более логических (булевых) переменных. Функция И для двух переменных А и В может быть записана в следующем виде:

$$У = A \cdot B = A \wedge B \qquad (I.I)$$

Можно сказать, что функция И для двух и более логических переменных истинна тогда и только тогда, когда все логические переменные являются истинными. Для случая двух переменных (I.I) значение У является истинным, когда обе переменные А и В истинны.

Комбинации переменных A, B и соответствующие им значения функции можно представить в виде таблицы истинности (табл. I)

Таблица I

Значение переменных		Значение функции
A	B	$y = A \wedge B$
0	0	0
0	I	0
I	0	0
I	I	I

СХЕМА ИЛИ

Схема ИЛИ – электронная схема, выполняющая операцию логического сложения (дизъюнкции) двух или более логических переменных. Операция ИЛИ для двух переменных A и B записывается следующим образом:

$$y = A + B = A \vee B \qquad (I.2)$$

На рис. I.3 изображена электрическая цепь с параллельно соединенными переключателями, которая выполняет операцию ИЛИ.

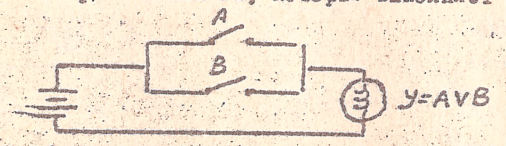


Рис. I.3

Индикаторная лампа светится, если замкнут один из переключателей A и B или оба.

Таблица истинности (табл. 2) отображает операцию ИЛИ.

Таблица 2

Значение переменных		Значение функции
A	B	$y = A \vee B$
0	0	0
0	I	I
I	0	I
I	I	I

Продолжение

Инструкция	Мнемокод	Интервал адресов в восьмеричном формате	Наименование	Время выполнения, мкс при выполнении	при игнорировании
Арифметические	CAC	0 - 377	Сложить арифметически слово	200	5
	BAC	0 - 377	Вычесть арифметически слово	200	5
	ПРС	0 - 7777	Параметр слова	5	5
	ПРК	0 - 4095_{10}	Параметр константы в десятичном формате	5	5
	НОП	без адреса	Нет операции (только выходная инструкция)	I20	5

Продолжение

Инструкция	Мнемокод	Интервал адресов в восьмеричном формате	Наименование	Время выполнения, мкс: при выполнении	при игнорировании
Для работы с данными	ЧБС		Читать байт старший	I20	5
	ПББ		Обменять содержимым байт старший и байт младший	I20	5
	СБС	0 - 377	Сброс слова	I20	5
	СЦС		Сдвиг циклический к старшим битам	I20	5
	СЦМ		Сдвиг циклический к младшим битам	I20	5
Сравнения	МШШ		Сравнить на "меньше"	I20	5
	ЕШШ	0 - 377	Сравнить на "больше"	I20	5
	РВН		Сравнить на "равно"	I20	5
	ПРН		Сравнить на "принадлежность пределам"	I20	5
Файловая	ПФФ	0 - 377	Переместить файл	переменная	5
	ЗСФ	0 - 377	Записать слово в файл	300	5
	ЧСФ	0 - 377	Читать слово из файла	300	5
	ПФМ	0 - I77	Переместить файл в модуль	переменная	5
	ПМФ	0 - I77	Переместить из модуля в файл	переменная	5
Экстренного обращения к модулям ввода-вывода	ЧТР	0 - I77	Читать из регистра	200	5
	ЗПР	0 - I77	Записать в регистр	200	5

СХЕМА НЕ (инвертор)

Инвертор - это электронная схема, выполняющая операцию НЕ над логической переменной.

Операцию НЕ для переменной А можно записать следующим образом:

$$N = \bar{A}$$

Таблица истинности (табл. 3) отображает операцию НЕ.

Таблица 3

А	$\bar{A}$
0	I
I	0

2. ОПИСАНИЕ ЯЗЫКА

2.I. Общие сведения

В контроллерах для запоминания инструкций, выполняющих логические операции, временную задержку, арифметическую обработку и т.д., служит программируемая память.

Совокупность необходимых инструкций или программа вводится в программируемую память пользователем при помощи прибора программирования и отладки МВ 96.03 (в дальнейшем - МВ96.03). Порядок работы МВ 96.03 описан в инструкции по эксплуатации гШЗ.048.023 ИЭ.

Исходное описание алгоритма управления объектом должно быть представлено на языке релейно-контактных схем или на языке алгебры Буля.

ПК сообщается с объектом управления с помощью модулей ввода-вывода, которые установлены в каркас ввода-вывода.
Вся информация, обрабатываемая процессором, сосредоточена в таблице данных (ТД).

В процессе работы ПК инструкции программы последовательно, начиная с первой, считываются процессором из памяти рабочей программы (ПРП) и выполняются, т.е. изменяется состояние ТД.
После выполнения последней инструкции происходит обмен информацией между определенной областью ТД с одной стороны и модулями

ввода-вывода с другой. Эти две фазы работы ПК - "счет" и "обмен" непрерывно циклически повторяются. Циклы работы процессора приведен на рис. 2.1. Первый цикл, после начального запуска, процессор проходит с результатом равным нулю. Это значит, что при первом проходе, независимо от результата вычислений функции, выходы, задействованные во всей программе, обнуляются и сбрасываются слова, на которых будут запускаться таймеры на включение.

Процессор содержит указатель адреса инструкций, содержимое которого увеличивается на I при выполнении очередной инструкции. При выполнении последней инструкции программы указатель адреса устанавливается в 0(адрес первой инструкции программы).

Процессор содержит два аккумулятора для накопления данных. Один одноразрядный аккумулятор для операций с логическими инструкциями и шестнадцатиразрядный для операций с многоразрядными инструкциями.

ПК может работать в трех режимах:

"Работа" (РАБ) - выполнение рабочей программы (РП);

"Программирование/Тестирование" (П/Т) - ввод и редактирование РП, формирование конфигурации памяти процессора, ввод констант в ТД, ввод буквенно-цифровых сообщений;

"Работа/Отладка" (Р/О) - выполнение РП, останов выполнения цикла РП, блокировка обмена с модулями ввода/вывода, снятие останова выполнения цикла, блокировки обмена с модулями ввода/вывода.

ПК имеет последовательный канал связи с МВ 96.03 и с Устройством программирования и отладки (МВ 96.02). Для ввода программы в ПК с помощью МВ 96.03 принята табличная форма записи.

Вводится программа по одной инструкции (строка таблицы).

Для МВ 96.02 принят язык релейно-контактных схем. Программа вводится по одной цепи релейно-контактной схемы.

Продолжение

Инструкция	Мнемокод	Интервал адресов в восьмеричном формате	Наименование	Время выполнения, мкс	
				при выполнении	при игнорировании
Для работы с подпрограммами	ВПП	0 - 7	Вызвать подпрограмму	I20	5
	НПП	0 - 7	Начало подпрограммы	I20	5
	КПП	77	Конец массива подпрограммы	I20	I20
Таймеров и счетчиков	ТОД		Таймер на отключение (0,I с)	250	5
	ТОС		Таймер на отключение (I с)	250	5
	ТВД		Таймер на включение (0,I с)	250	5
	ТВС		Таймер на включение (I с)	250	5
	ТНД		Таймер накапливающий (0,I с)	250	5
	ТНС		Таймер накапливающий (I с)	250	5
	СЧВ	0 - 377	Счетчик на вычитание	250	5
	СЧС		Счетчик на сложение	250	5
	СЧЦ		Счетчик циклов	250	5
Для работы с данными	ЧТС		Читать слово	I20	5
	ЗПС		Записать слово	I20	5
	ПУС		Поразрядно умножить слово	I20	5
	ПСС	0 - 377	Поразрядно сравнить слово	I20	5
	ЗБМ		Записать байт младший	I20	5
	ЗБС		Записать байт старший	I20	5
	ЧБМ		Читать байт младший	I20	5

ПРИЛОЖЕНИЕ

СПИСОК ИНСТРУКЦИЙ ЯЗЫКА СИМВОЛИЧЕСКОГО КОДИРОВАНИЯ

Инструкция	Мнемокод	Интервал адресов в восьмеричном формате	Наименование	Время выполнения, мкс	
				при выполнении	при игнорировании
Логическая	-II-	0 - 37717	Прямой опрос входа	7	5
	-И-	0 - 37717	Инверсный опрос входа	7	5
	-()-	0 - 37717	Включить выход	9	5
	-(S)-	0 - 37717	Включить выход и зафиксировать	9	5
	-(R)-	0 - 37717	Выключить выход и зафиксировать	9	5
	⊢	без адреса	Начало одного ветвления	7	5
	⊣	То же	Конец одного ветвления	7	5
	↙	-"-	Перенос маркера в начало ветвления	7	5
Структурирования	НСТ	0 - 7	Начало сегмента	120	120
	ЗСТ	0 - 7	Закрыть сегмент	120	5
	ОСТ	0 - 7	Открыть сегмент	120	5
	ССТ	0 - 7	Сбросить сегмент	120	5
	НСТ	77	Конец цикловой программы	4000	4000
	ЗСТ	77	Программный останов	120	5
	НБЛ	0 - 76	Начало блока	120	5
	НБЛ	77	Возврат из подпрограммы	120	5
	ПБЛ	0 - 76	Перейти на блок	120	5
	УИС	0 - 17	Установить номер сектора ТД	120	5

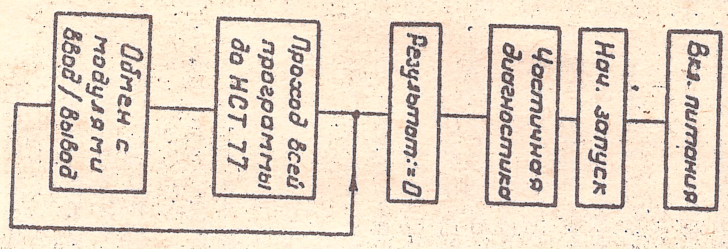


Рис. 2.1

Цикл работы процессора

2.2. Структура памяти ПК

Вся программируемая память МС 59.06 имеет объем 2000 слов, а МС 59.07 имеет объем памяти 4000 слов. Программируемая память состоит из ТД, защищаемой области, области задания маски обмена, области задания конфигурации, области задания текущего состояния, области диагностики отказов и сбоев. Структура памяти приведена на рис. 2.2.

Объем ТД и ПРП задается пользователем в процессе программирования: Объем ТД и ПРП не может превышать для МС 59.06 -3717_8 слов, а для МС 59.07 - 7717_8 слов. Программируемая память в общем случае может состоять из 8 секторов (00,01 ... 07) - для МС 59.06 и из 16 секторов (00,01 ... 17) - для МС 59.07.

Объем одного сектора составляет 400_8 слов (рис. 2.3). В пределах одного сектора адреса слов от 000 до 377_8.

Процессор может читать информацию из ТД по четырехзначному либо по трехзначному адресу в восьмеричном формате согласно списку инструкций языка ПК приведенному в приложении.

2.2.1. Таблица данных

Минимальной информационной единицей, обрабатываемой процессором, является бит. Бит принимает два возможных логических значения "0" либо "1". Канал модуля ввода или вывода дискретных сигналов имеет отображение в ТД в виде бита. Формирование адреса бита приведено на рис. 2.4.

Упорядоченная совокупность из шестнадцати бит образует слово. Биты в слове нумеруются двухзначными восьмеричными числами 00, 01 ... 17. Младшие восемь бит слова образуют нулевой, а старшие - первый байт слова.

Обрабатываемые программой потребителя данные (слова и биты) располагаются в ТД.

В ТД находится информация о состоянии модулей ввода-вывода. В фазе обмена процессор производит обмен информацией между дискретными модулями ввода-вывода и ТД. Схема обмена между ТД и модулями ввода-вывода приведена на рис. 2.5.

В ТД в словах, свободных от обмена между модулями ввода-вывода и ТД, могут быть организованы таймеры, счетчики, в свободные слова можно записывать биты промежуточной информации, различные константы. Примеры возможного распределения памяти, таблицы для распределения модулей ввода-вывода, побитное распределение ТД, пословное распределение приведены на рис. 2.6, 2.7, 2.8, 2.9.

Продолжение

Номер инструкции	Мнемокод	Примечание	Номер инструкции	Мнемокод	Примечание
0074	-|/|- 05000		0109	-||- 05000	
0075	-||- 05001		0110	-||- 05001	
0076	-||- 05002		0111	-|/|- 05002	
0077	-|/|- 05003		0112	-||- 05003	
0078	-()- 05106		0113	-()- 05115	
0079	-|/|- 05000		0114	-||- 05000	
0080	-||- 05001		0115	-||- 05001	
0081	-||- 05002		0116	-||- 05002	
0082	-||- 05003		0117	-|/|- 05003	
0083	-()- 05107		0118	-()- 05116	
0084	-||- 05000		0119	-||- 05000	
0085	-|/|- 05001		0120	-||- 05001	
0056	-|/|- 05002		0121	-||- 05002	
0087	-|/|- 05003		0122	-||- 05003	
0088	-( )- 05110		0123	-()- 05117	
0089	-||- 05000		0124	НБЛ 77	
0090	-|/|- 05001		0125	НПП 77	
0091	-|/|- 05002				
0092	-||- 05003				
0093	-( )- 05111				
0094	-||- 05000				
0095	-|/|- 05001				
0096	-||- 05002				
0097	-|/|- 05003				
0098	-()- 05112				
0099	-||- 05000				
0100	-|/|- 05001				
0101	-||- 05002				
0102	-||- 05003				
0103	-()- 05113				
0104	-||- 05000				
0105	-||- 05001				
0106	-|/|- 05002				
0107	-|/|- 05003				
0108	-()- 05114				

Продолжение

Номер инструкции	Мнемокод	Примечание
7	САС 201	
8	ЧЧ 020 02	
9	САС 202	
10	ЧЧ 020 03	
11	САС 203	
12	ЧЧ 020 04	
13	САС 204	
14	ЧЧ 020 05	
15	САС 205	
16	ЧЧ 020 06	
17	САС 206	
18	ЧЧ 020 07	
19	САС 207	
20	ЧЧ 020 10	
21	САС 210	
22	ЧЧ 020 11	
23	САС 211	
24	ЧЧ 020 12	
25	САС 212	
26	ЧЧ 020 13	
27	САС 213	
28	ЧЧ 000 00	
29	↙	
30	ЧЧ 000 00	
31	ЗПС 220	
32	НСТ. 77	
33	НПП 77	

Общая структура памяти

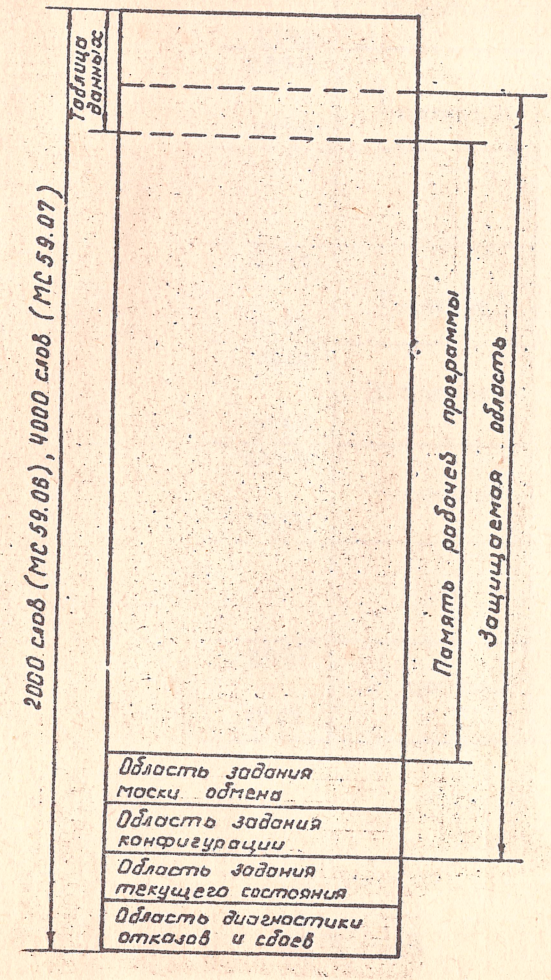


Рис. 2.2

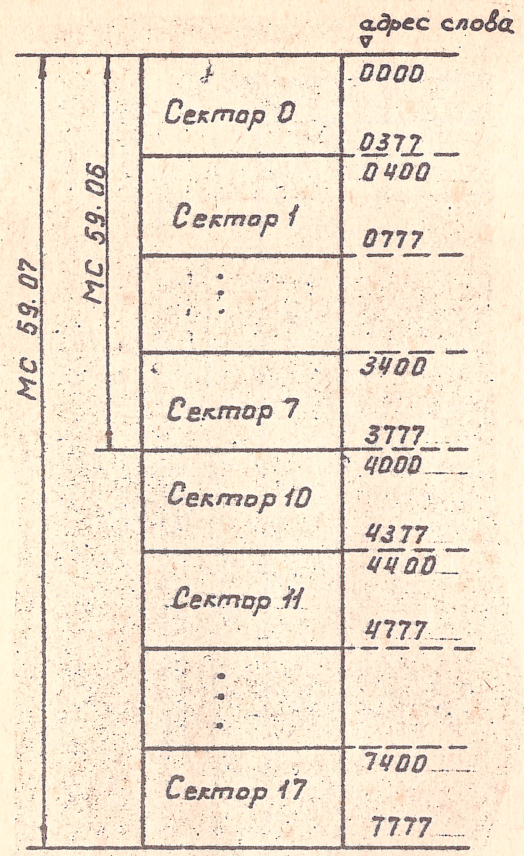


Рис. 2.3

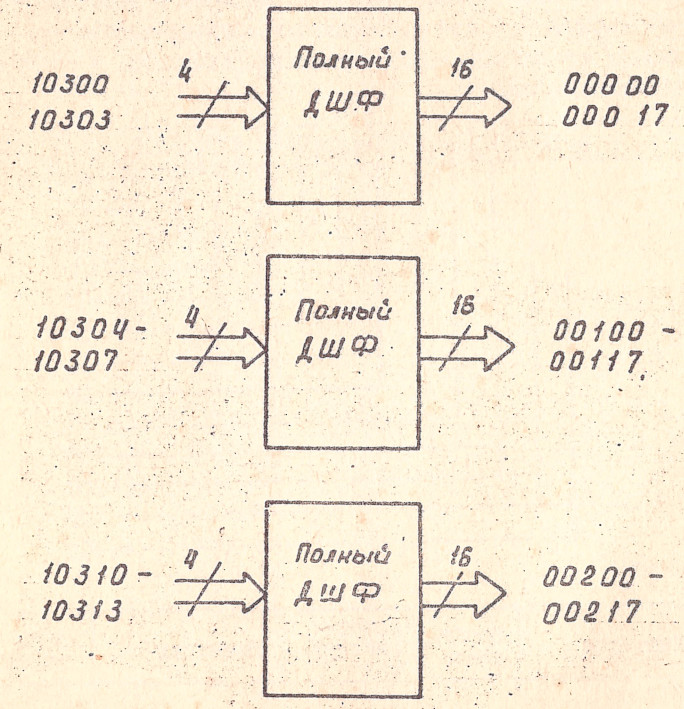
Программа.

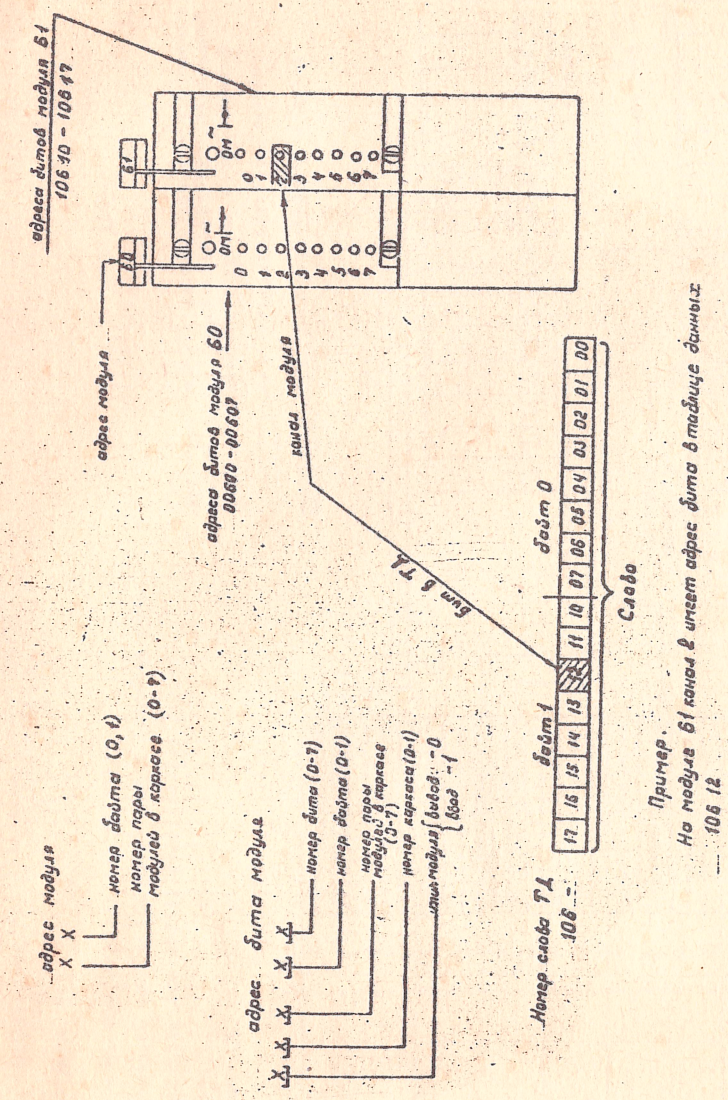
Номер инструкции	Мнемокод	Примечание	Номер инструкции	Мнемокод	Примечание
0000	НСТ 00		0037	⊣⊢ 05004	
0001	НБЛ 00		0038	ВПЯ 00	
0002	-(S)- 05004		0039	⊣⊢ 05004	
0003	⊣⊢ 10300		0040	ЧТС 051	
0004	-( )- 05000		0041	ЗПС 002	
0005	⊣⊢ 10301		0042	НСТ 77	
0006	-( )- 05001		0043	НПП 00	
0007	⊣⊢ 10302		0044	⊣/⊢ 05000	
0008	-( )- 05002		0045	⊣/⊢ 05001	
0009	⊣⊢ 10303		0046	⊣/⊢ 05002	
0010	-( )- 05003		0047	⊣/⊢ 05003	
0011	⊣⊢ 05004		0048	-( )- 05100	
0012	ВПП 00		0049	⊣/⊢ 05000	
0013	⊣⊢ 05004		0050	⊣/⊢ 05001	
0014	ЧТС 051		0051	⊣/⊢ 05002	
0015	ЗПС 000		0052	⊣⊢ 05003	
0016	⊣⊢ 10304		0053	-( )- 05101	
0017	-( )- 05000		0054	⊣/⊢ 05000	
0018	⊣⊢ 10305		0055	⊣/⊢ 05001	
0019	-( )- 05001		0056	⊣⊢ 05002	
0020	⊣⊢ 10306		0057	⊣/⊢ 05003	
0021	-( )- 05002		0058	-( )- 05102	
0022	⊣⊢ 10307		0059	⊣/⊢ 05000	
0023	-( )- 05003		0060	⊣/⊢ 05001	
0024	⊣⊢ 05004		0061	⊣⊢ 05002	
0025	ВПП 00		0062	⊣⊢ 05003	
0026	⊣⊢ 05004		0063	-( )- 05103	
0027	ЧТС 051		0064	⊣/⊢ 05000	
0028	ЗПС 001		0065	⊣⊢ 05001	
0029	⊣⊢ 10310		0066	⊣/⊢ 05002	
0030	-( )- 05000		0067	⊣/⊢ 05003	
0031	⊣⊢ 10311		0068	-( )- 05104	
0032	-( )- 05001		0069	⊣/⊢ 05000	
0033	⊣⊢ 10312		0070	⊣⊢ 05001	
0034	-( )- 05002		0071	⊣/⊢ 05002	
0035	⊣⊢ 10313		0072	⊣⊢ 05003	
0036	-( )- 05003		0073	-( )- 05105	

Пример реализации дешифратора на четыре входа и шестнадцать выходов

Входная информация : 05000 - 05003

Выходная информация : 05100 - 05117.





Пример.
На модуле 61 канал 8 имеет адрес бита в таблице данных — 106 12

Рис. 2.4

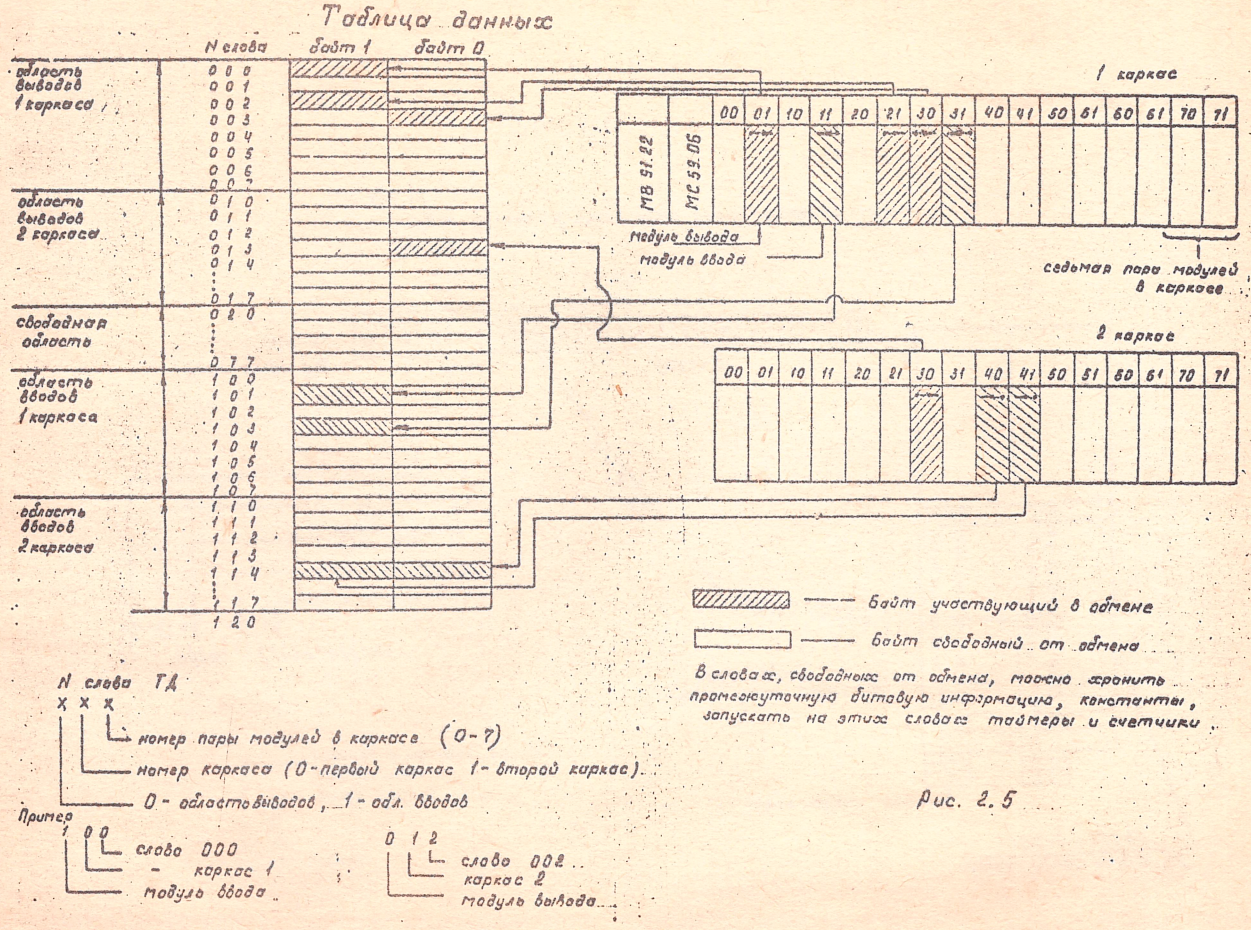
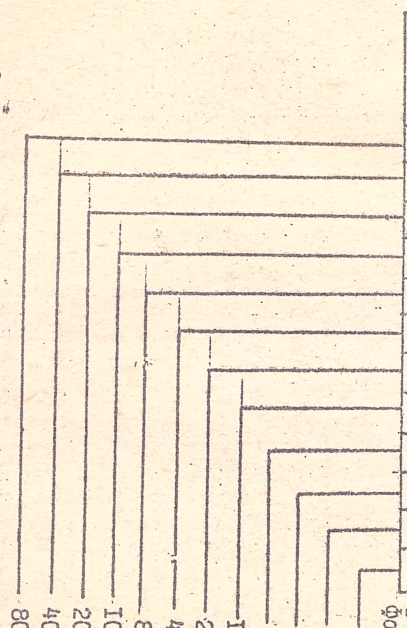


Рис. 2.5

```
СБС 050
-I/I- X3
-()- У1
-()- У2
ШБП 00
НСТ 77
НШП 77
```

X - условное обозначение дискретного входа
У - условное обозначение дискретного выхода

Программа перевода двоично-десятичного числа в двоичный код. Исходные данные: в слове 020 хранится двоично-десятичное число (три тетрады), в словах 200 - 213 - весовые константы, в слове - 220 - результат (двоичный код).



Программа

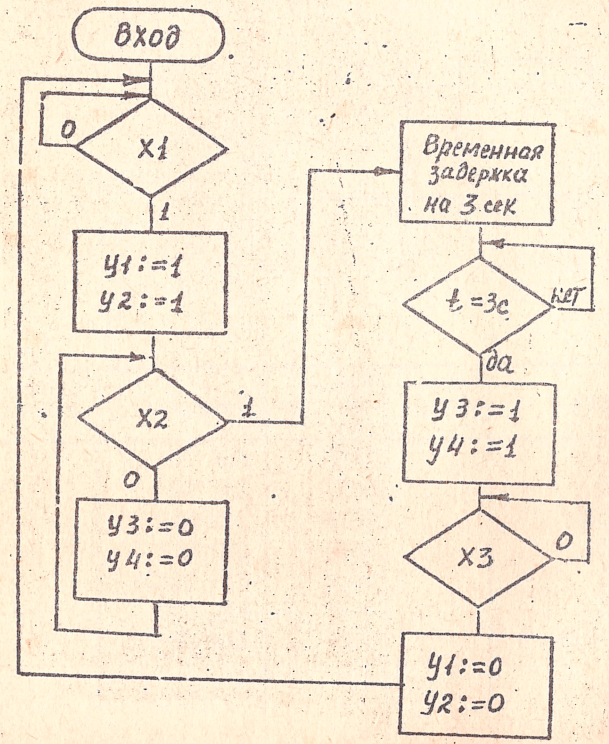
Номер инструкции	Мнемокод	Примечание	Номер инструкции	Мнемокод	Примечание
0000	НСТ 00				
0001	НБП 00				
0002	СБС 220				
0003	ЧТС 220				
0004	-II- 020 00				
0005	САС 200				
0006	-II- 020 01				

Продолжение

Номер инструкции	Мнемокод	Примечание	Номер инструкции	Мнемокод	Примечание
0007	-()- 00000		0023	НСТ 77	
0008	ЧСФ 050		0024	НПП 77	
0009	ПРС А				
0010	ПРС W1				
0011	ПРК N				
0012	ЧСФ 051				
0013	ПРС В				
0014	ПРС W2				
0015	ПРК N				

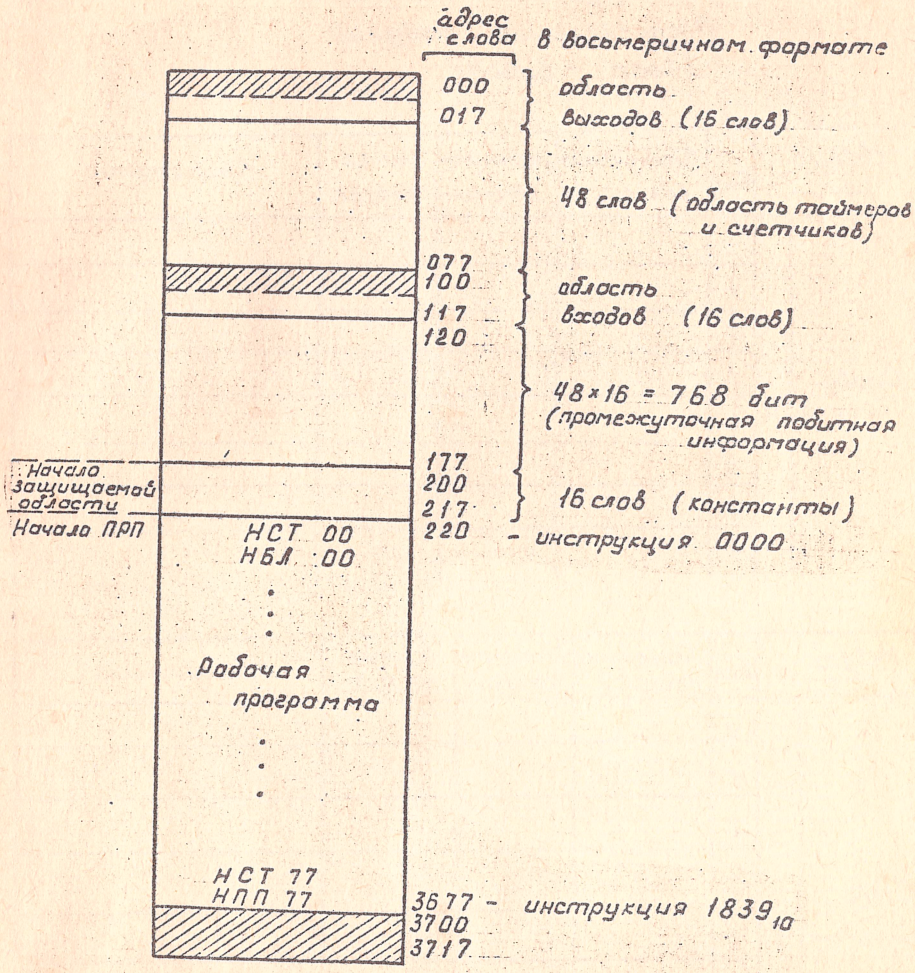
Пример реализации алгоритма, представленного блок-схемой.

Алгоритм



Программа

```
--  НСТ 00
    НБЛ 00
    -II- XI
    ПБЛ 01
    НБЛ 01
    -()- У1
    -()- У2
    -II- Х2
    ПБЛ 02
    -()- У3
    -() - У4
    НБЛ 02
    -I/I- 0501б
    ТВД 050
    ПРК 0030
    -II- 0501б
    ПБЛ 03
    НБЛ 03
    -()- У3
    -()- У4
    -II- Х3
    ПБЛ 04
    НБЛ 04
```

Рабочая программа, в которой инструкций побитной обработки - 1800, таймеров - 40, выполняется ≈ за 26 мс.

Расчет:
1800 × 7 мкс (время выполнения побитной инструкции) ≈ 12 мс
40 × 250 мкс (время выполнения инструкций таймеров) ≈ 10 мс
Обмен ≈ 4 мс.

Рис. 2.6

Программируемый контроллер
Распределение модулей
ввода - вывода

Модуль		Бит	Обозначение сигналов	Элементы схемы	Номер провода	Примечание
Слово ⌴⌴⌴	0	0				
		1				
		2				
		3				
		4				
		5				
		6				
		7				

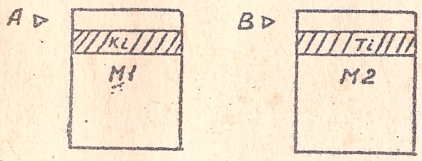
Модуль		Бит	Обозначение сигналов	Элементы схемы	Номер провода	Примечание
Слово ⌴⌴⌴	1	0				
		1				
		2				
		3				
		4				
		5				
		6				
		7				

Рис. 2.7

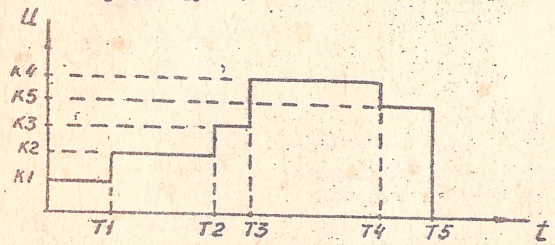
Реализация временной функции.

$$U = f(t),$$

Исходная информация: массив М1 – набор констант К i и М2 – констант Т i.



A – начальный адрес массива М1,
B – начальный адрес массива М2
W1, W2 – адреса слов, N – число констант в массиве.
График функции



Программа

Номер инструкции	Мнемокод	Примечание	Номер инструкции	Мнемокод	Примечание
0000	НСТ 00		0016	ПБЛ 02	
0001	НБЛ 00		0017	НБЛ 02	
0002	СБС 050		0018	ТВД 052	
0003	СБС 051		0019	ПРС W2	
0004	ПБЛ 01		0020	-II- 0521б	
0005	НБЛ 01		0021	СБС 052	
0006	-I/I- 000 00		0022	ПБЛ 01	

Программа

Номер инструкции	Мнемокод	Примечание	Номер инструкции	Мнемокод	Примечание
0000	НСТ 00		0036	НСТ 02	ручной
0001	НБЛ 00		0037	НБЛ 00	режим
0002	-II- I0I0I	Пуск авт.	0038	-I/I- I0I 03	
0003	-I/I- I0I00	реж.	0039	-II- I0I 06	А -
0004	ОСТ 0I		0040	-( )- 00002	вперед
0005	ЗСТ 02		0041	-I/I- I0I02	
0006	ПБЛ 0I		0042	-II- I0I07	А -
0007	-II- I0I0I		0043	-( )-00003	назад
0008	-II- I0I00		0044	-I/I- I0I05	
0009	ОСТ 02		0045	-II- I0II0	В -
0010	ЗСТ 0I		0046	-( )- 00005	вперед
0011	ПБЛ 0I		0047	-I/I- I0I04	
0012	НБЛ 0I		0048	-II- I0III	В -
0013	-I/I- I0I0I		0049	-( )- 00006	назад
0014	ПБЛ 00		0050	НСТ 77	
0015	НСТ 0I	авт.реж.	0051	НПП 77	
0016	НБЛ 00				
0017	-I/I- I0I02	Приведение			
0018	-()- 000 03	в исходное			
0019	-I/I- I0I 04	состояние			
0020	-()- 00006				
0021	-II- I0I 02				
0022	-II- I0I04				
0023	ПБЛ 0I				
0024	НБЛ 0I	движения А вперед			
0025	-I/I- I0I03				
0026	-()- 00002				
0027	-II- I0I03				
0028	ПБЛ 02				
0029	НБЛ 02	А-назад			
0030	-I/I- I0I02	В-вперед			
0031	-()- 00003				
0032	-I/I- I0I05				
0033	-()- 00005				
0034	-II- I0I05				
0035	ПБЛ 00				

Программируемый контроллер _____

Побитное распределение таблицы данных

Процессор _____

Объем ТД _____

Слова от ___ до ___

Слово	Бит	Описание	Слово	Бит	Описание
	0 0			0 0	
	0 1			0 1	
	0 2			0 2	
	0 3			0 3	
	0 4			0 4	
	0 5			0 5	
	0 6			0 6	
	0 7			0 7	
	1 0			1 0	
	1 1			1 1	
	1 2			1 2	
	1 3			1 3	
	1 4			1 4	
	1 5			1 5	
	1 6			1 6	
	1 7			1 7	
	0 0			0 0	
	0 1			0 1	
	0 2			0 2	
	0 3			0 3	
	0 4			0 4	
	0 5			0 5	
	0 6			0 6	
	0 7			0 7	
	1 0			1 0	
	1 1			1 1	
	1 2			1 2	
	1 3			1 3	
	1 4			1 4	
	1 5			1 5	
	1 6			1 6	
	1 7			1 7	

Комментарий _____

Рис. 2.8

Программируемый контроллер _____
Пословное распределение
таблицы данных

Процессор _____
Объем ТД _____
Слова от ____ до ____

Слово		Описание	
		Байт старший	Байт младший
	0		
	1		
	2		
	3		
	4		
	5		
	6		
	7		
	0		
	1		
	2		
	3		
	4		
	5		
	6		
	7		
	0		
	1		
	2		
	3		
	4		
	5		
	6		
	7		
	0		
	1		
	2		
	3		
	4		
	5		
	6		
	7		

Комментарий _____

Рис. 2.9

Модуль IO	Байт	Бит	Обозначение сигналов	Элементы схемы	Номер провода	Примечание
Слово IOI	0	4	X4			
		5	X5			
		6	X6			
		7	X7			

Модуль II	Байт	Бит	Обозначение сигналов	Элементы схемы	Номер провода	Примечание
Слово IOI	I	0	XIO			
		I	XII			
		2				
		3				
		4				
		5				
		6				
		7				

Модификация конфигурации ПК.

С помощью МВ 96.03 надо ввести:

НАЧАЛО 30 0120

НАЧАЛО ПРП 0120

ВРЕМЯ ЦИКЛА 0377

В/В КАРКАС I

У--XX----------

В/В КАРКАС 2

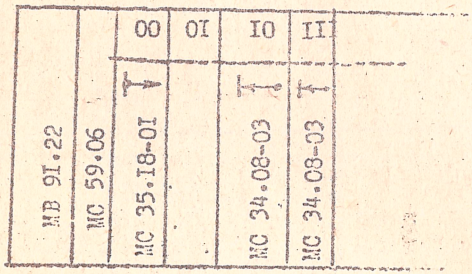
Предполагается два режима работы: ручной и автоматический.

Автоматический режим.

Исходное состояние: тележки А и В находятся над датчиками Х2, Х4.

Работа: тележка А движется от датчика Х2 к Х3, при подходе к датчику Х3 начинается движение тележки В от Х4 к Х5, а тележка А движется от Х3 к Х2. После подхода к датчику Х5, тележка В движется к датчику Х4.

Каркас ПК:



Распределение модулей ввода-вывода

Модуль 00	Байт	Бит	Обозначение сигналов	Элементы схемы	Номер провода	Примечание
Слово 000	0	0	-			
		1	-			
		2	У2			
		3	У3			
		4	-			
		5	У5			
		6	У6			
		7	-			

Модуль 10	Байт	Бит	Обозначение сигналов	Элементы схемы	Номер провода	Примечание
Слово 101		0	Х0			Автомат. реж.
		1	Х1			Пуск
		2	Х2			
		3	Х3			

2.2.2. Область задания маски обмена (рис. 2.10).

Маска обмена – это информация о наличии дискретного модуля в каркасе и разрешение на обмен информацией с этим модулем.

Маску обмена можно задавать, пользуясь специальной функцией МВ 96.03.

Соответствие бита слова маски обмена и номера модуля в каркасе приведено на рис. 2.11.

2.2.3. Область задания конфигурации

Под конфигурацией системы понимается начало защищаемой области, начало рабочей программы и время цикла. Эти три параметра пользователю необходимо ввести с помощью МВ 96.03.

Часть памяти потребителя, называемая защищаемой областью, может храниться, по выбору потребителя, в ОЗУ с подпиткой от резервного источника питания либо в ППЗУ с ультрафиолетовым способом стирания. Содержимое защищаемой части памяти может быть выведено на магнитную ленту или введено с магнитной ленты. Минимальный адрес, с которого может начинаться защищаемая область, равен 120_8 (номер слова ТД). С этого же адреса может начинаться и ПРП.

Для МС 59.06 максимально возможная ПРП занимает объем от 120_8 до 3677_8 слов или имеет от 0 до 1903_{10} инструкций. Для МС 59.07 максимально возможная ПРП занимает объем от 120_8 до 7677_8 слов или имеет от 0 до 3951_{10} инструкций.

Максимальное время цикла – 426 мс соответствует коду 0377_8.

Превышение времени цикла фиксируется в системных отказах (отказ ПВЦ). Таблица соответствия времени цикла – коду, приведена на рис. 2.12.

2.2.4. Область задания текущего состояния.

Процессор при прохождении рабочей программы пользуется информацией области задания текущего состояния, и при определенных условиях ее изменяет. Структура области задания текущего состояния приведена на рис. 2.13. При выполнении инструкций структурирования информация этой области изменяется.

2.2.5. Область диагностики отказов и сбоев.

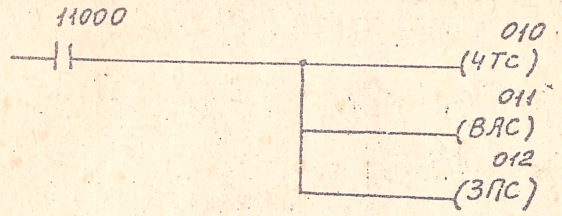
Доступ в область отказов и сбоев дает использование инструкции УНС. На рис. 2.14, рис. 2.15, рис. 2.16 раскрыто содержимое слов данной области.

Бит в любом разряде означает отказ, ноль-норму.

Область задания маски обмена		МС59.06	МС59.07
Маска обмена выходов I-го каркаса		3700	(7700)
Маска обмена выходов 2-го каркаса		3701	(7701)
Маска обмена входов I-го каркаса		3702	(7702)
Маска обмена входов 2-го каркаса		3703	(7703)
Область задания конфигурации			
Начало защищаемой области		3704	(7704)
Начало ПРП		3705	(7705)
	время цикла	3706	(7706)
Контрольная сумма		3707	(7707)
Область задания текущего состояния			
S1	S0	3710	(7710)
S3	S2	3711	(7711)
S5	S4	3712	(7712)
S7	S6	3713	(7713)
Область диагностики отказов и сбоев			
Системные отказы		3714	(7714)
и сбои		3715	(7715)
Отказы I-го каркаса		3716	(7716)
Отказы 2-го каркаса		3717	(7717)

Рис. 2.10

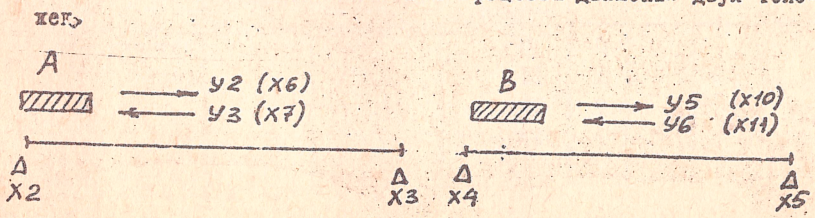
Номер слова	Содержимое в десятичном формате	
010	2048	исходная
011	0032	информация
012	2016	результат



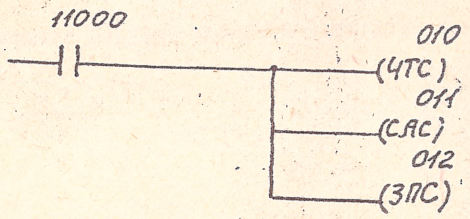
Программа

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
1	НБЛ 00	
2	-II- II0 00	
3	ЧТС 010	
4	ВАС 011	
5	ЗПС 012	
6	НСТ 77	
7	НПП 77	

2.14. Примеры по составлению рабочих программ.
Реализация технологического процесса движения двух теле-



Номер слова	Содержимое в десятичном формате	
010	2048	} исходные данные
011	0032	
012	2080	- результат



Программа:

```
    НСТ 00
    НБЛ 06
-II- IIO 00
    ЧТС 010
    САС 011
    ЗПС 012
    НСТ 77
    НПП 77
```

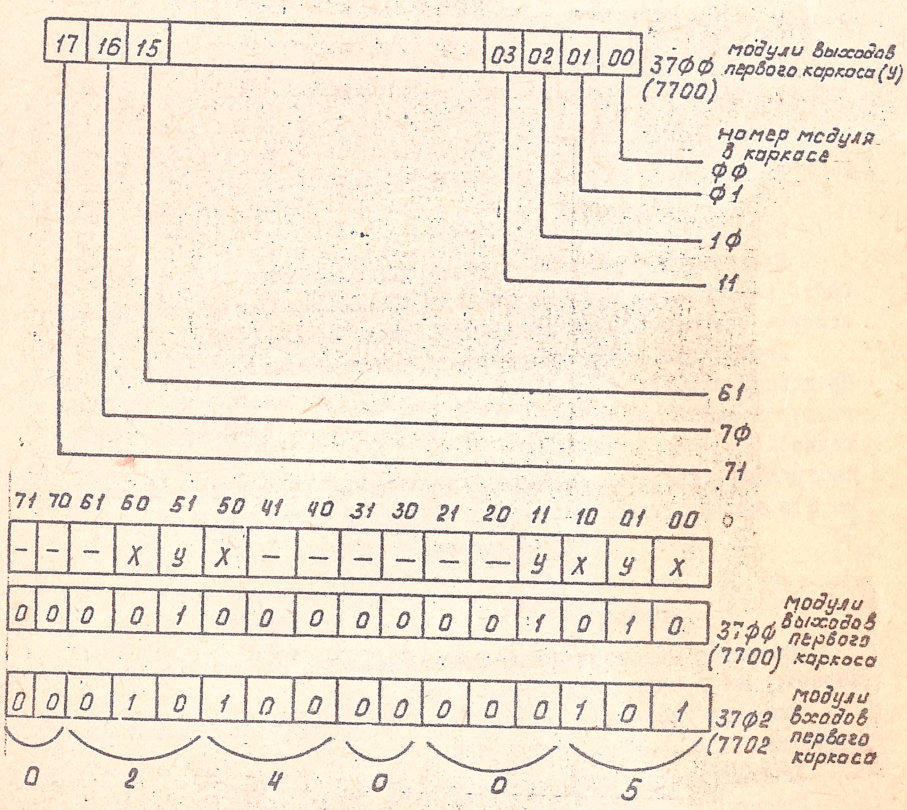
ВЫЧЕСТЬ АРИФМЕТИЧЕСКИ СЛОВО
ВАС ппп

Когда цепь, в которой находится инструкция вычитания, становится истинной, производится вычитание числа, находящегося в аккумуляторе из числа записанного по адресу слова. Если в результате вычитания получено отрицательное число, то включается бит I4.

Пример.

Содержимое слов (010) и (011) равно соответственно 2048 и 32. Необходимо вычесть 32 из 2048 и результат записать по адресу (012).

Соответствие бита слова маски обмена и номера модуля в каркасе



Модули выходов - У,
Модули входов - Х.

В слово 3700 надо записать константу в восьмеричном формате 010012, в 3701 - 000 000, в 3702 - 024005, в 3703 - 000 000.

Рис. 2.11

Время цикла (мс)	Код в восьмеричном формате
I0	0006
30	0022
50	0036
70	0052
90	0066
II0	0I02
I30	0II0
I50	0I23

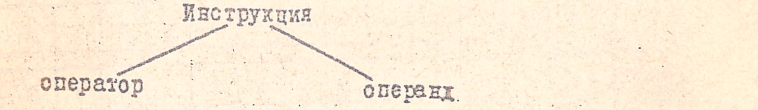
Рис. 2.I2.

2.3. Структура программы

Программа представляет собой упорядоченную последовательность инструкций, т.е. каждая инструкция имеет свой порядковый номер - четырехзначное десятичное число 0000, 000I ... и т.д.

Инструкция - это распоряжение, выполняемое процессором. Процессор выполняет не любые инструкции, а только те, которые входят в набор инструкций, заложенных в его память (см. приложение).

Инструкция является наименьшей самостоятельной единицей программы и в общем случае состоит из оператора и операнда.



Оператор описывает выполнение процессором функции. Она указывает, что должен делать процессор.

Операнд указывает данные, необходимые для выполнения инструкции.

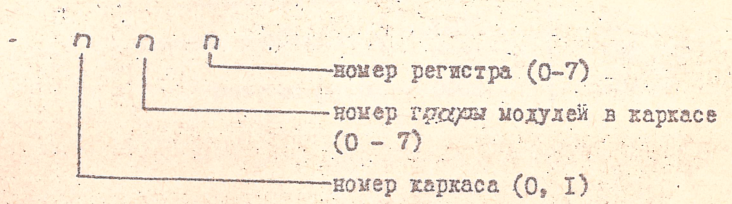
Структура программы приведена на рис. 2.I7.

Программа должна начинаться инструкциями НСТ 00 - начало сегмента 0, НБЛ 00 - начало блока 0 и заканчивается: НСТ 77 - конец цикловой программы, НПП 77 - конец массива подпрограмм.

Пример бланка для написания программы приведен на рис. 2.I8.

Программа подразделяется на сегменты, начало каждого из ко-

Структура адреса регистра



Распределение регистров в группе модулей следующее:

	N0	N1
номер регистров	0	4
	I	5
	2	6
	3	7

N - номер пары модулей в каркасе

2.I3. Арифметические инструкции

Эти инструкции дают возможность арифметической обработки данных в пользовательской программе. С помощью арифметических инструкций можно обрабатывать целые положительные числа в пределах от 0 до 4095. Бит I4 в служебных разрядах включается, если в результате арифметических операций результат вышел за указанные пределы. Арифметические операции - условные.

СЛОЖИТЬ АРИФМЕТИЧЕСКИ СЛОВО

САС п п п

адрес слова

Когда цепь, в которой находится инструкция сложения становится истинной, производится операция сложения чисел, находящихся в аккумуляторе и по адресу слова, следующего за САС. Если в результате сложения получено число больше чем 4095, то включается бит I4 в слове, где записан результат.

Пример.

Содержимое слов (0I0) и 0II) равно соответственно 2048 и 32. Эти числа необходимо сложить и результат поместить по адресу (0I2).

ПЕРЕМЕСТИТЬ ФАЙЛ В МОДУЛЬ

ПРН XXX

ПРС XXXX – начальный адрес памяти модуля

ПРС XXXX – начальный адрес ТД

ПРК XXXX – длина файла L

ПРН X X X

- X — номер регистра
- X — номер группы модулей в каркасе
- X — номер каркаса

При выполнении этой инструкции файл из ТД перемещается в память модуля по указанному адресу за один скан, если запускающее выражение находится в состоянии "I".

ПЕРЕМЕСТИТЬ ИЗ МОДУЛЯ В ФАЙЛ

ПМФ X X X

ПРС X X X X – начальный адрес памяти модуля

ПРС X X X X – начальный адрес ТД

ПРК(К) X X X X – длина файла

При выполнении этой инструкции файл из памяти модуля перемещается в ТД по указанному адресу.

Программирование отдельных модулей, входящих в состав ПК, приведено в технических описаниях или в инструкциях по эксплуатации на модули.

2.12. Инструкции экстренного обращения к модулям ввода-вывода

ЧИТАТЬ ИЗ РЕГИСТРА

ЧТР nnn

При выполнении этой инструкции содержимое регистра модуля пересылается в аккумулятор.

ЗАПИСАТЬ В РЕГИСТР

ЗПР nnn

При выполнении этой инструкции содержимое аккумулятора ТД пересылается в регистр модуля.

Инструкции ЧТР и ЗПР применяются при работе с модулями, имеющими регистры.

Область задания текущего состояния

	17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0
3710 (7710)	сегмент 01								сегмент 00							
3711 (7711)	03								02							
3712 (7712)	05								04							
3713 (7713)	07								06							

7	6	5	4	3	2	1	0

Номер активного блока

0 0 — обнуление

0 1 — игнорирование

1 1 — вычисление

Рис. 2.13

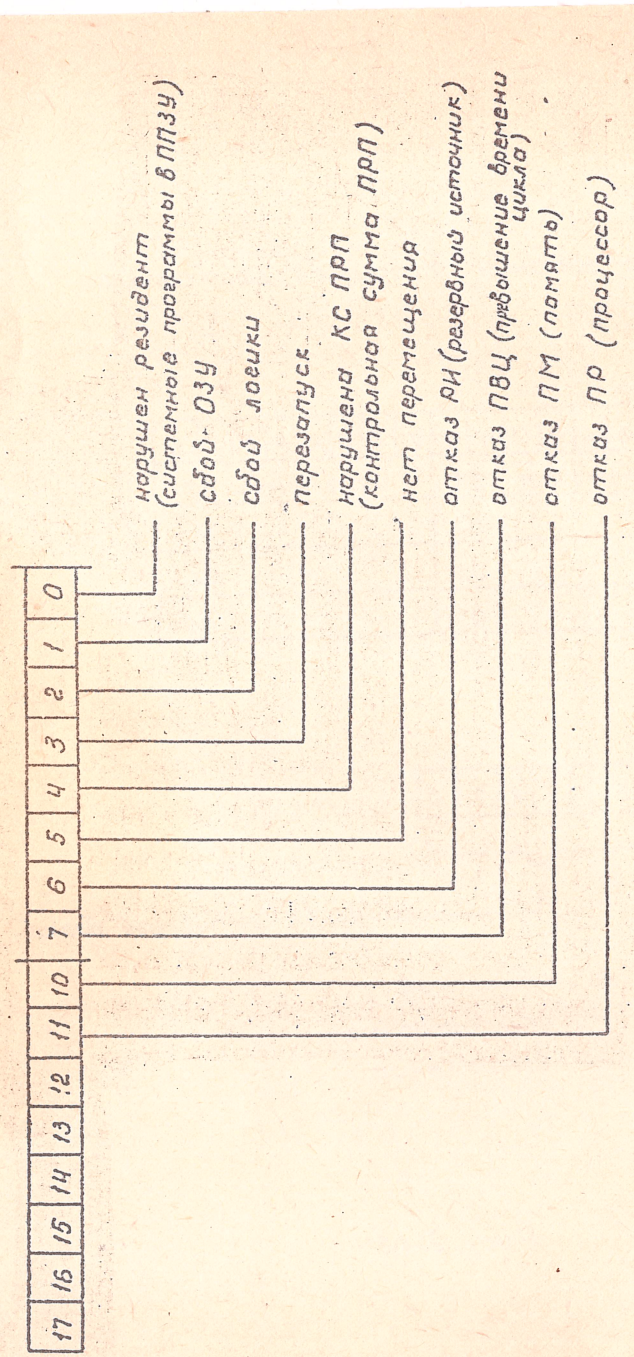


Рис. 2.14

ЗАПИСАТЬ СЛОВО В ФАЙЛ
ЗСФ XXX - адрес счетчика
ПРС XXXX - начальный адрес файла
ПРС XXXX - адрес слова
ПРК XXXX - длина файла

При изменении запускающего выражения из "0" в "1" содержимое по адресу слова переписывается в начальный адрес файла. При следующем изменении запускающего выражения из "0" в "1" содержимое по адресу слова записывается в следующий адрес файла, т.е. начальный адрес файла + 1.

Пример.

Записать в ТД константы:

Номер слова	константы	
0036	123 123	} исходные данные
0037	177 177	

Программа

Номер инструкции	Мнемокод	Примечание	Номер инструкции	Мнемокод	Примечание
0001	НСТ 00		0013	-II- 030 16	
2	НБЛ 00		14	ЗСФ 031	
3	СБС 031		15	ПРС 0040	
4	СБС 032		16	ПРС 0037	
5	ОСТ 01		17	ПРК 0020	
6	ПБЛ 01		18	-II- 0311 4	
7	НБЛ 01		19	-()- 0000	
8	-I/I- 030 16		20	-II- 0311 6	
9	ТВД 030		21	ПБЛ 02	
10	ПРК 0010		22	НБЛ 02	
11	НСТ 01		23	ЧТС 036	
12	НБЛ 00		24	ЗПС 037	
			25	СБС 051	
			26	НБЛ 00	
			27	НСТ 77	
			28	НПП 77	

Продолжение

Номер слова	Содержимое в восьмеричном формате
I43	000 003
I44	000 004
I45	000 005
000	377 377

Программа

Номер инструкции	Мнемокод	Примечание	Номер инструкции	Мнемокод	Примечание
0000	НСТ 00		0008	ПРК 0010	
0001	НБЛ 00		0009	НСТ 01	
0002	СБС 031		000а	НБЛ 00	
0003	ОСТ 01		0011	-II- 032 I6	
0004	ПБЛ 01		0012	ЧСФ 031	
0005	НБЛ 01		0013	ПРС 0140	
0006	-I/I- 032 I6		0014	ПРС 0000	
0007	ТВД 032		0015	ПРК 0006	
			0016	-II- 0314	бит вкл. при наличии ошибки
			0017	-()- 0214	бит вкл. при наличии ошибки
			0018	НСТ 77	
			0019	НПП 77	

После включения программы:

Время с начала выполнения программы	Информация в слове 000
I с	000 000
2 с	000 001
3 с	000 002
4 с	000 003
5 с	000 004
6 с	000 005

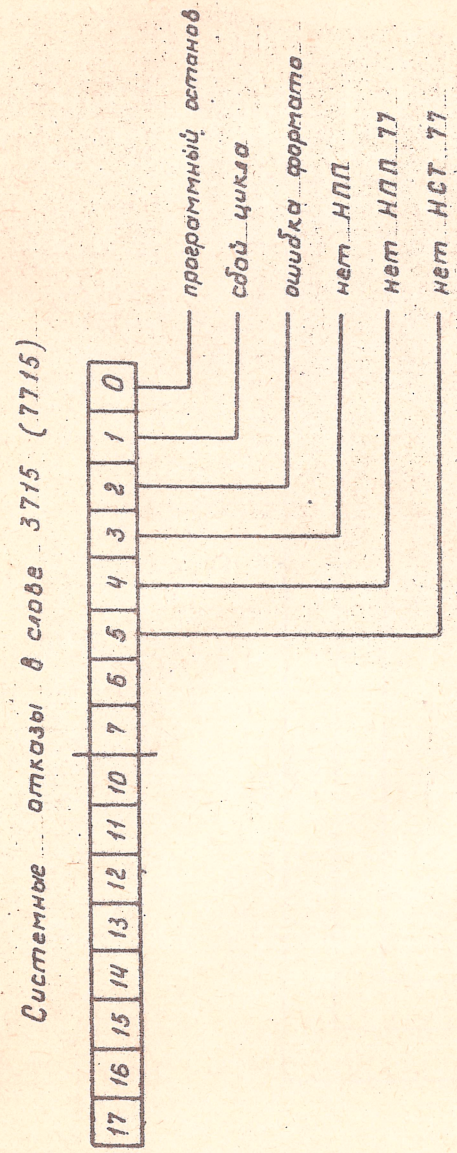


Рис. 2.15

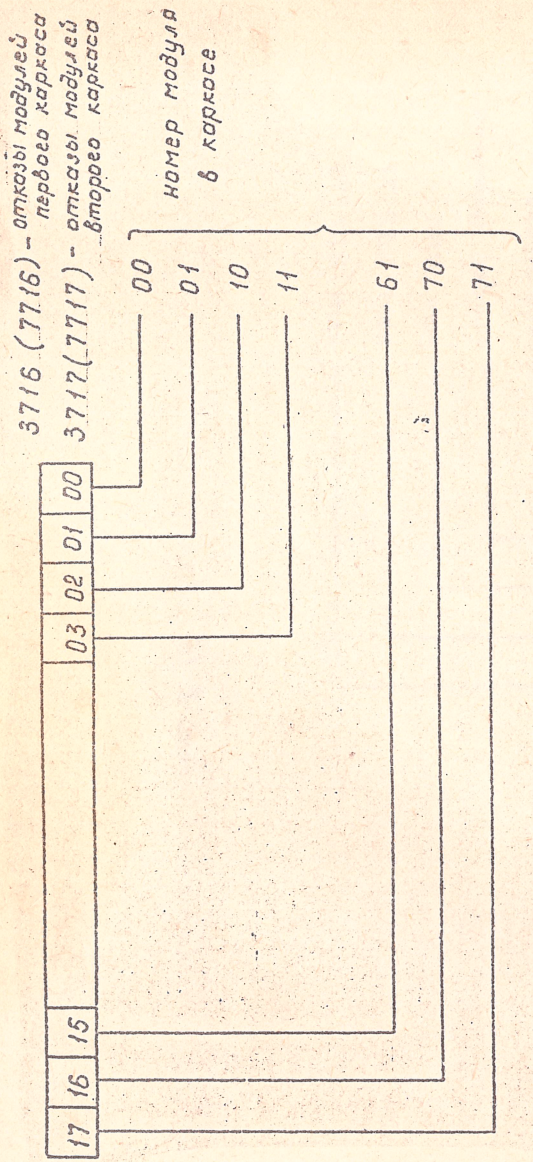


Рис. 2.16

Номер инструкции	Мнемокод	Примечание
0008	ПРК 0020	
9	НСТ 01	
10	НБЛ 00	
11	-II- 0III6	- через каждые 2с бит принимает
12		истинное значение
13	ПФФ 010	
14	ПРС 0030	начало массива А
15	ПРС 0100	начало массива В
16	ПРК 0008	количество пересылаемых слов
17	ПРК 0001	- одно слово за один скан
18	-II- 01014	
19	ЗСТ 77	
20	-II- 010 16	
21	-()- 000 02	- бит возбудится через 16с
22	НСТ 77	
23	НПП 77	

Перемещение информации из первого файла во второй происходит по одному слову через каждые 2 с.

ЧИТАТЬ СЛОВО ИЗ ФАЙЛА

ЧСФ XXX - адрес счетчика

ПРС XXXX - начальный адрес файла

ПРС XXXX - адрес слова

ПРК XXXX - длина файла

При изменении состояния запускающего выражения из "0" в "1" происходит перезапись содержимого слова файла по адресу слова. При следующем изменении запускающего выражения из "0" в "1" содержимое следующего слова файла переписывается по тому же адресу слова.

Пример.

Исходная информация:

Номер слова	Содержимое в восьмеричном формате
140	000 000
141	000 001
142	000 002

где . L – длина файла, равная количеству слов файла ;
V – количество слов за скан, равное скорости перемещения (Рис. 2.23) .

Выполнение инструкции осуществляется пересылкой количества слов за один скан, равного скорости перемещения. Перемещение производится слово в слово. Содержимое счетчика будет увеличиваться после каждого перемещения, а перемещение происходит при изменении запускающего выражения из 0 в I. Как только содержимое счетчика станет равным длина файла L , бит I6, характеризующий выполнение инструкции, возбудится.

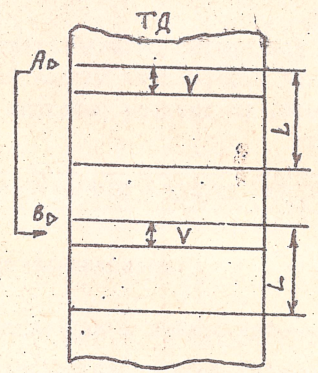


Рис. 2.23

Пример.
Программа

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
1	НБЛ 00	
2	ОСТ 01	
3	СБС 010	
4	ПБЛ 01	
5	НБЛ 01	
6	-I/I- 011 I6	
7	ТВД 011	

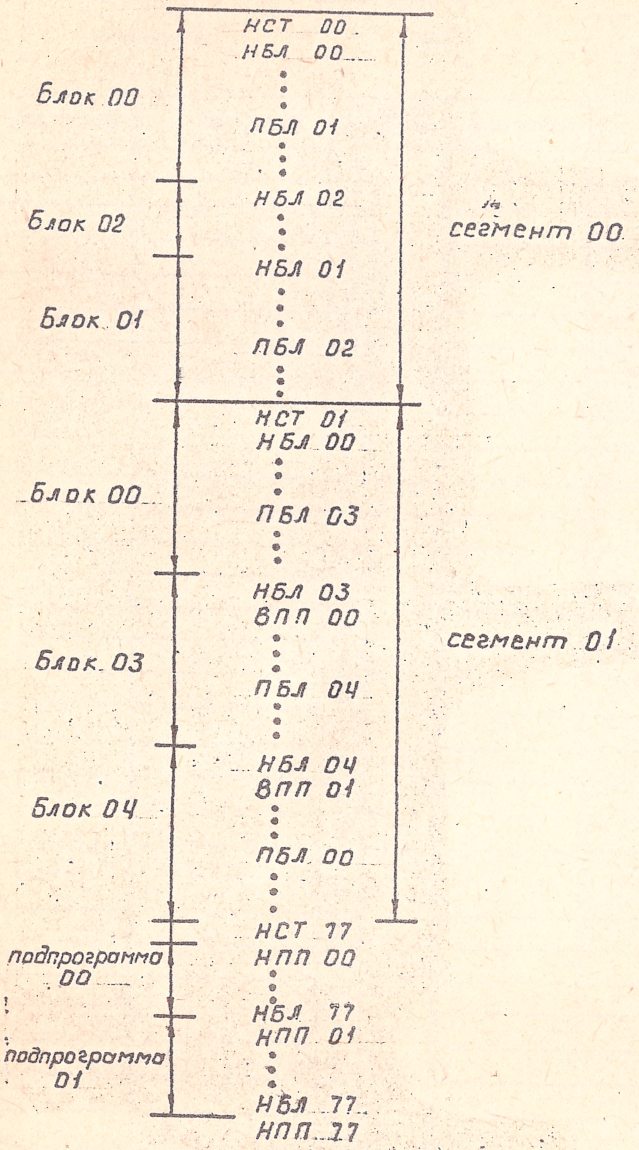


Рис. 2.17

Программируемый контроллер _____

Программа _____

Лист _____

Номер инструкции	Мнемокод	Примечание	Номер инструкции	Мнемокод	Примечание
0			0		
1			1		
2			2		
3			3		
4			4		
5			5		
6			6		
7			7		
8			8		
9			9		
0			0		
1			1		
2			2		
3			3		
4			4		
5			5		
6			6		
7			7		
8			8		
9			9		
0			0		
1			1		
2			2		
3			3		
4			4		
5			5		
6			6		
7			7		
8			8		
9			9		

Рис. 2.18

Продолжение

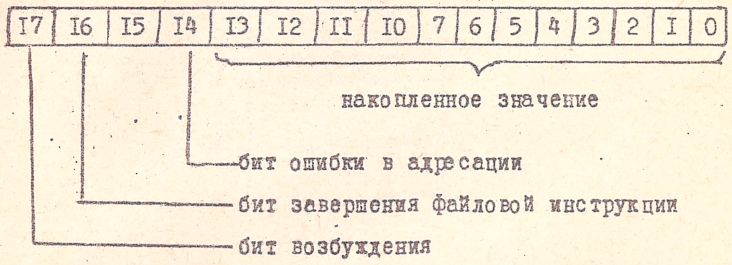
Номер инструкции	Мнемокод	Примечание
00 I7	-(-)-000 03	
I8	-И- 000 03	
I9	-( )-000 02	
20	НСТ 77	
2I	НПП 77	

В результате выполнения программы, через 2с включится бит 000 02.

2.II. Файловые инструкции

Файловые инструкции - это такие инструкции, которые оперируют с группами слов ТД.

Файл определяется как группа последовательных слов ТД, используемых для хранения информации. Каждая файловая инструкция использует счетчик. Слово, на котором работает счетчик файловой инструкции имеет формат:



Если запускающее выражение равно 0 и взведен I6 бит, то счетчик обнуляется. Все файловые инструкции условные. Состояния файловых инструкций: возбуждение, завершение и ошибка отображаются соответственно служебными битами I7, I6, I4 слова счетчика, связанного с данной файловой инструкцией.

ПЕРЕМЕСТИТЬ ФАЙЛ

ПФФ *nnn* , где *nnn* - адрес счетчика.

Перемещаемый файл задается параметрами:

ПФФ X X X - адрес счетчика ;

ПРС X X X X - начальный адрес файла A;

ПРС X X XX - начальный адрес файла B;

ПРК X X X X - длина файла *L* ;

ПРК X X X X - скорость перемещения *V*,

Пример.

Необходимо сравнить содержимое по адресу 010 с содержимым по адресу 011, если равны, то включить 000 00 бит, если не равны, то сравнить на "меньше". При истинности инструкции, включить бит 000 01, при ложном состоянии выполнить выдержку времени 2 с, прочесть содержимое по адресу 010 и сравнить на принадлежность пределам 4095 и 1000. При истинном результате, включить бит 000 03.

При ложном результате - включить бит 000 02.

Исходные данные:

Номер слова	Содержимое в десятичном формате
010	2048
011	4095
012	1000

Программа:

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
1	НБЛ 00	
2	РВН 010	
3	ПРС 0011	
4	-()- 000 00	
5	МНШ 010	
6	ПРС 0012	
7	-()- 000 01	если равно и больше бит не включается
8	-И- 000 01	
9	ТВС 003	
10	ПРК 0002	
11	-II- 003 16	
12	НБЛ 01	
13	НБЛ 01	
14	ПРН 010	
15	ПРС 0011	
16	ПРС 0012	

торых отмечается инструкцией "начало сегмента".

В свою очередь каждый сегмент программы состоит из блоков, начало каждого из которых отмечается инструкцией "начало блока". Максимально возможное число различных сегментов программы равно 8 (от 0 до 7) и максимально возможное число различных блоков в одном сегменте равно - 76_8. Максимально возможное число подпрограмм равно восьми.

Количество сегментов в программе, количество блоков в сегменте, а также порядок следования блоков внутри сегмента могут быть произвольными. В частном случае, весь сегмент может быть одним блоком, а вся программа - одним сегментом. Программа может состоять из одного сегмента и одного блока.

Каждый сегмент (за исключением нулевого сегмента) может находиться в одном из трех состояний; вычисление, игнорирование, обнуление.

Вычисление - значит программа активного блока выполняется.

Игнорирование - программа сегмента не выполняется.

Обнуление - процессор проходит инструкции сегмента с результатом равны нулю в течение одного скана.

Нулевой сегмент всегда вычисляется (при вкл. питания этот сегмент обнуляется).

При первом проходе программы все сегменты находятся в обнулении - это значит, что процессор проходит программу с результатом равным нулю и поэтому все выходы, включенные в программу, обнуляются, а слова, на которых запущены таймеры на включение - сбрасываются. После команды НСТ 77, все сегменты устанавливаются в режим игнорирования.

В следующих проходах процессор при переходе от сегмента к сегменту пользуется информацией области меток активных блоков (область задания текущего состояния на рис. 2-13). В эту область заносится информация, после каждой команды структурирования.

Информация о текущем состоянии сегмента занимает 1 байт.

В шесть младших разрядов (0-5) записывается номер активного блока, который начнет работать в следующем скане.

В шестой и седьмой разряды записываются биты информации, после выполнения инструкций структурирования: ОСТ, ЗСТ, ССТ.

Каждый блок сегмента может находиться в одном из двух состояний: активном или пассивном. На протяжении одного цикла сканирования программы в каждом сегменте может быть только один активный блок. При включении питания ПК во всех сегментах программы устанавливаются активными нулевые блоки.

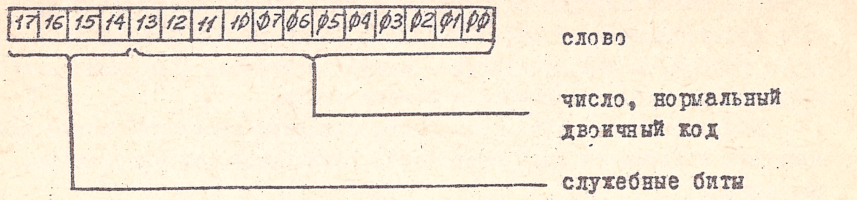
2.4. Представление числа

Содержимое слова (т.е. комбинация состояний его бит) может подвергаться арифметической обработке. В этом случае оно имеет следующий формат:

биты 00, 01 – 13 – число в нормальном двоичном коде;

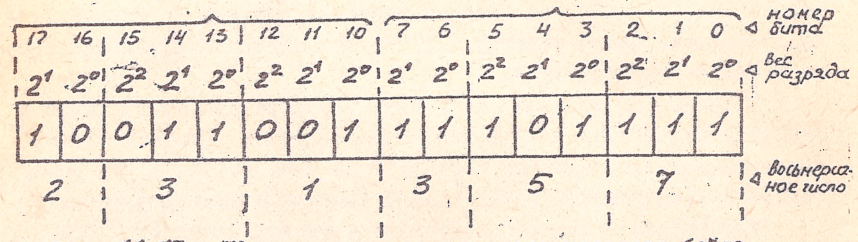
биты 14–17 – служебные.

Состояние служебных битов приобретает различное значение в зависимости от выполняемой инструкции:



Таким образом, в контроллере допускается непосредственная обработка чисел в интервале от 0 до 4095. 4095 – максимальное (десятичное) значение уставки таймера. ПК обрабатывает только положительные числа.

Содержимое одного слова можно также закодировать восьмеричным числом:



Биты ØØ–Ø7 содержат восьмеричное число младшего байта.

Биты 1Ø–17 содержат восьмеричное число старшего байта.

2.5. Инструкции логические

Эта группа инструкций используется для опроса состояния битов ТД – входов или выходов, служебных битов таймеров, счетчиков, а также для управления состоянием битов выходов.

Логические инструкции подразделяются на адресные и безадресные.

2.10. Инструкции сравнения

Эта группа инструкций позволяет производить сравнения в рабочей программе. Инструкции работают с данными в формате целых чисел и в пределах от 0 до 4095.

Инструкции сравнения являются входными инструкциями. Если цепь содержит только инструкцию сравнения и выходную инструкцию, то при истинности инструкции сравнения выходная инструкция включается, при ложном состоянии инструкции сравнения, несохраняемая выходная инструкция выключается. Если цепь содержит другие входные инструкции, кроме сравнения, то выходная инструкция включается, если можно отыскать хотя бы один замкнутый путь из логически истинных инструкций. Если же такого пути нет, то несохраняемая выходная инструкция выключается, а сохраняемая – остается в прежнем состоянии.

СРАВНИТЬ НА РАВНО

РВН *nnn* ,РВН nnn

ПРС *mmmm* ПРК kkkk

Если содержимое по адресу слова *nnn* равно содержимому по адресу *mmmm*, то инструкция логически истинна, если же данные не равны, то инструкция ложна. Если ПРК–то сравнение с константой.

СРАВНИТЬ НА ПРИНАДЛЕЖНОСТЬ ПРЕДЕЛАМ

ПРН *nnn* , ПРН nnn

ПРС *mmmm* ПРК m m m m

ПРС *kkkk* ПРК kkkk

Если число, хранящееся по адресу nnn одновременно больше или равно числу, записанному по адресу kkkk и меньше или равно числу, записанному по адресу m m m m, то инструкция ПРН истинна. Если после инструкции–ПРК, то сравнение в том же порядке, только с константами.

СРАВНИТЬ НА МЕНЬШЕ

МНШ *nnn* , МНШ nnn

ПРС *mmmm* ПРК kkkk.

Если число, хранящееся по адресу *nnn* меньше, чем число записанное по адресу *mmmm* , то инструкция МНШ истинна, в противном случае – логически ложна.

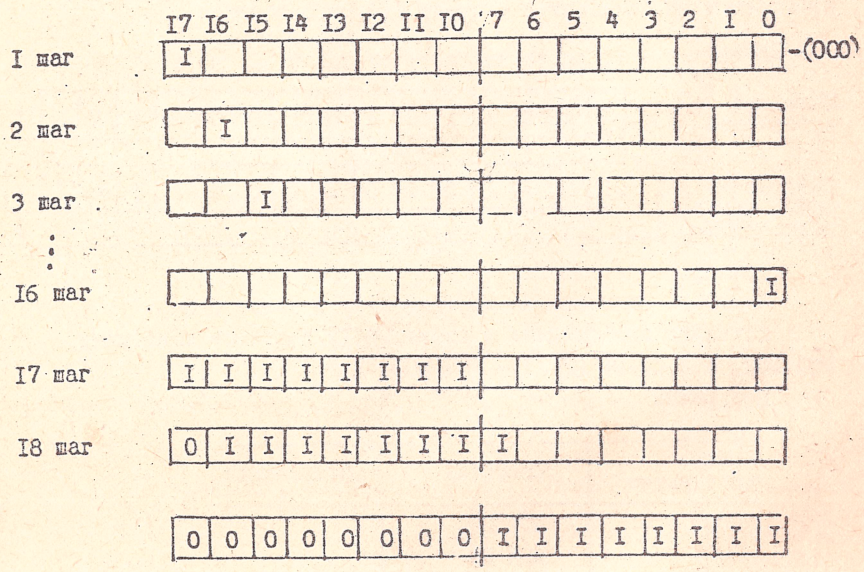
СРАВНИТЬ НА БОЛЬШЕ

БЛШ *nnn* , БЛШ nnn

ПРС *mmmm* , ПРК kkkk.

Если число, хранящееся по адресу *nnn* больше, чем число, записанное по адресу *mmmm*, то инструкция БЛШ логически истинна, в противном случае –ложна. Инструкции сравнения с ПРК работают только в МС 59.07.

Номер инструкции	Мнемокод	Примечание
0005	НБЛ 01	
0006	-I/I- 003 I6	
0007	ТВД 003	
0008	ПРК 0010	
0009	-II- 003 I6	
0010	СЦМ 000	
0011	-II- 000 00	
0012	ПБЛ 02	
0013	НБЛ 02	
0014	ЧТС 012	
0015	ЗПС 000	
0016	ПБЛ 01	
0017	НСТ 77	
0018	НШП 77	



Адрес бита должен находиться в пределах 000 00 - 377 $I7_8$.

Структура адреса бита - см. рис. 2.4.

Операторы адресных инструкций на языке релейно-контактных символов:

-II- - прямой опрос входа;

-И- - инверсный опрос входа;

-()- - включить выход;

-(S)- - включить выход и зафиксировать;

-(R)- - выключить выход и зафиксировать;

Операторы безадресных инструкций:

⊏ - начало одного ветвления;

⊐ - конец одного ветвления;

⊏ - перенести маркер в начало ветвления.

Операторы адресных инструкций на языке булевой алгебры:

, - конъюнкция;

./ - конъюнкция с инверсией;

= - присвоить;

=S - присвоить "I" и зафиксировать;

=R - присвоить "0" и зафиксировать.

Операторы безадресных инструкций:

(- скобка открывающая ;

) - скобка закрывающая ;

+ - дизъюнкция.

ПРЯМОЙ ОПРОС ВХОДА

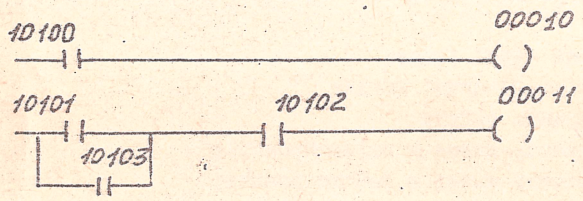
Когда входное устройство находится в замкнутом состоянии, как например, включенный конечный выключатель, индицируется включенное состояние. Это состояние передается в соответствующий бит памяти. Когда в программе встречается инструкция опроса на включение с этим адресом, процессор определяет, что входное устройство включено и считает инструкцию истинной. Если входное устройство размыкается, процессор считает эту инструкцию ложной.

Если цепь содержит только эту инструкцию и выходную инструкцию, то выходная инструкция возбуждается (бит включается), когда входная инструкция истинна.

Если же входная инструкция ложна (вход разомкнут), то нефиксированный выход выключится.

Если в цепи содержатся еще входные инструкции, то выходная инструкция возбуждается, когда может быть найден по крайней мере

один замкнутый путь из логически истинных инструкций, ведущий от шины питания к выходной инструкции. Если такого пути не существует, то незафиксированные выходы отключаются. Пример двух цепей:

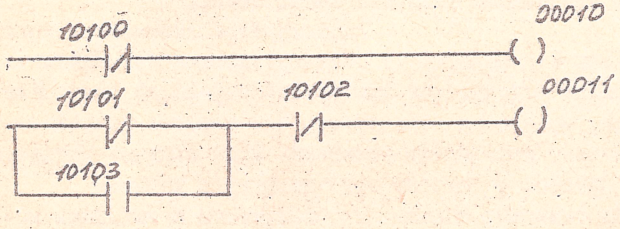


Выход 000 10 возбудится, если контакт 101 00 замкнут. Выход 000 11 возбудится, если контакты 101 01 и 101 02 замкнуты, либо замкнуты контакты 101 03 и 101 02.

ИНВЕРСНЫЙ ОПРОС ВХОДА

Если входное устройство не замкнуто, то входной замок, к которому это устройство присоединено, индицирует выключенное состояние. Когда в программе встречается инструкция инверсного опроса, процессор определяет, что входное устройство выключено и считает инструкцию истинной. Если входное устройство замыкается, то процессор считает инструкцию ложной.

Если цепь содержит только инструкцию инверсного опроса и выходную инструкцию, то выходная инструкция включается, когда инструкция инверсного опроса находится в истинном состоянии, т.е. вход отключен. Если инструкция ложна, то несохраняемый выход отключается. Пример двух цепей:



Выход 000 10 включится, если контакт 101 00 разомкнут.

Выход 000 11 включится, если контакты 101 01 и 101 02 разомкнуты, либо 101 03 замкнут, а 101 02 разомкнут.

бит информации переходит на место младшего бита.

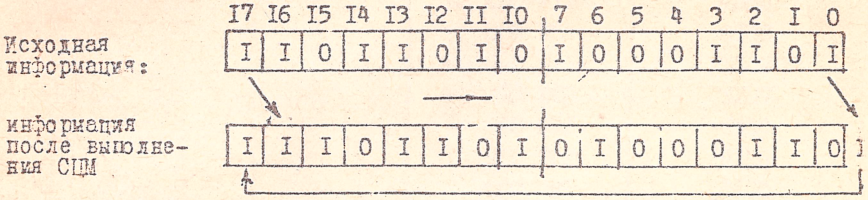
Затем, все оставшиеся биты в содержимом сдвигаются влево или вправо на один разряд.

СДВИГ ЦИКЛИЧЕСКИЙ К МЛАДШИМ БИТАМ

СЦМ n n n

При выполнении инструкции СЦМ содержимое по адресу слова сдвигается на один бит в сторону младшего разряда.

Пример.

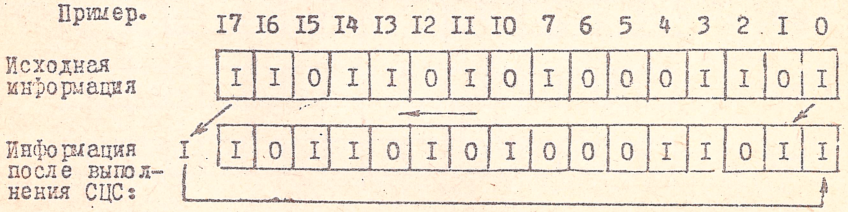


СДВИГ ЦИКЛИЧЕСКИЙ К СТАРШИМ БИТАМ

СЦС n n n

При выполнении инструкции СЦС содержимое по адресу слова сдвигается на один бит в сторону старшего разряда.

Пример.



Пример.

Через 1с включать поочередно биты в слове 010.

Исходная информация: в слово (011) записать константу $200\ 000_8$ и в слово (012) - $377\ 000_8$

Программа

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
0001	НБЛ 00	
0002	ЧТС 011	
0003	ЗПС 000	
0004	НБЛ 01	

Исходная информация

Номер слова	Содержимое в восьмеричном формате
010	111 111
011	222 222
012	333 333
013	000 000

Программа

```
НСТ 00
НБЛ 00
ЧБМ 010
ЗБМ 012
ЧБС 011
ЗБС 012
ЧТС 012
ЗПС 013
ПББ 013
НСТ 77
КПП 77
```

Информация, после выполнения программы:

Номер слова	Содержимое в восьмеричном формате
010	111 111
011	222 222
012	222 111
013	111 222

Инструкция сдвига бита

При помощи этих инструкций можно организовать битовый синхронный сдвиг внутри шестнадцатиразрядного слова.

Инструкции сдвига бита являются выходными. Когда цепь переходит из ложного в истинное состояние, инструкция начинает работать.

Инструкции действуют следующим образом.
Сначала один бит выводится из сдвигаемого содержимого по адресу слова. Если сдвиг в сторону младшего разряда, то бит, переходит на место стершего. При сдвиге в сторону старшего разряда, старший

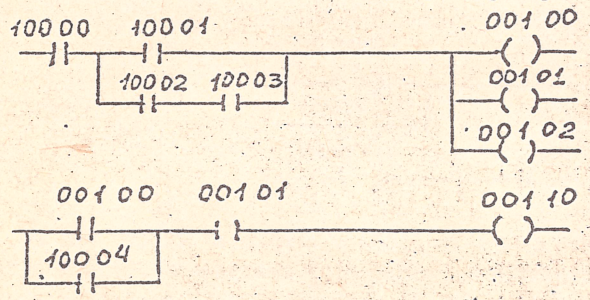
ВКЛЮЧЕНИЕ ВЫХОДА

Инструкция включения выхода используется для включения отдельных битов памяти. Если адрес бита соответствует адресу зажима выходного модуля, то выходное устройство, соединенное с этим зажимом, также включается. Состояние включения или выключения выходного бита определяется логическим состоянием цепи в рабочей программе пользователя.

Если в цепи существует замкнутый путь из логически истинных входных инструкций, то выходная инструкция возбуждается (включается). Если такого пути нет, то условия цепи ложны и выходная инструкция выключается. Если условия цепи ложны, то выходное устройство отключается.

Инструкция включения выхода подобна катушке реле. Выходная инструкция управляется предшествующими ей входными инструкциями в программной цепи. Катушка реле управляется контактами в электрической цепи. Замкнутый путь из логически истинных инструкций соответствует электрической цепи из замкнутых электрических контактов.

Любой бит ТД в пределах 0- 377_8 слов может быть опрошен любой инструкцией опроса столько раз, сколько необходимо. Таким образом, бит управляемый выходной инструкцией, также может быть опрошен необходимое количество раз, например:



Биты 00100 и 00101 в первой цепи - выходные. Эти же биты во второй цепи участвуют во входных инструкциях.

ФИКСАЦИЯ И РАСФИКСАЦИЯ ВЫХОДА

Инструкции фиксации и расфиксации выхода являются сохраняемыми выходными инструкциями. Они обычно используются в паре для управления битом ТД. Если адрес инструкции фиксации соответствует адресу зажима выходного модуля, то выходное устройство, ссе-

диненное с этим зажимом включается при включении бита, что определяется условиями его цепи.

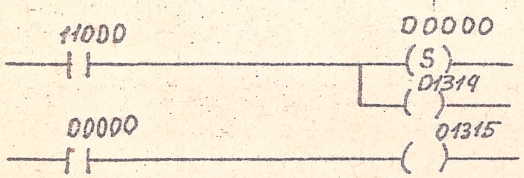
Если в цепи может быть найден замкнутый путь из логически истинных инструкций, то инструкция фиксации выхода включает управляемый ею бит. Когда условия в цепи перестают быть истинными, бит, управляемый инструкцией фиксации остается включенным, если он перед этим был включен.

Инструкция расфиксации, имеющая тот же адрес, что и инструкция фиксации, используется для выключения бита памяти. Когда в цепи устанавливается замкнутый путь из логически истинных инструкций, инструкция расфиксации выключает управляемый ею бит.

Когда ПК переходит из режима РАБОТА в режим ПРОГРАММИРОВАНИЯ, или происходит потеря питания, последняя из включенных инструкций фиксации или расфиксации сохраняет состояние управляемого ею бита. При переходе в режим РАБОТА выходное устройство, соответствующее зафиксированному биту, сразу включается, даже если условия в цепи ложны.

Программа пользователя может опрашивать бит, управляемый инструкциями фиксации и расфиксации, произвольное количество раз.

Например, включение катушки с фиксацией:



При замыкании контакта II000 в память по адресу 000 00 записывается логическая I, зажигаются светодиоды 01314 и 01315. При размыкании контакта II0 00 светодиод 01314 гаснет, а светодиод 01315 продолжает гореть. Если произойдет потеря питания, а затем его восстановление, светодиод 01315 не гаснет, то есть логическая единица по адресу 000 00 сохраняется в памяти.

Программа для реализации цепи с фиксированной катушкой:
Перед вводом программы необходимо задать:30,ПРП,время цыкла и указать места расположения модулей. Номер инструкции пользователю вводить не нужно.Вводится только мнемокод.

ЧТС 012
ПСС 013
ЗПС 014
НСТ 77
НПП 77

Номер слова	Содержимое в восьмеричном формате	
012	324 017	Исходная
013	123 226	информация
014	170 146	- результат

ЧИТАТЬ БАЙТ МЛАДШИЙ

ЧБМ ппп

При выполнении инструкции ЧБМ младший байт содержимого по адресу слова пересылается в младший байт аккумулятора.

ЧИТАТЬ БАЙТ СТАРШИЙ

ЧБС ппп

При выполнении инструкции ЧБС старший байт содержимого по адресу слова пересылается в старший байт аккумулятора.

ЗАПИСАТЬ БАЙТ МЛАДШИЙ

ЗБМ ппп

При выполнении этой инструкции содержимое младшего байта аккумулятора пересылается в младший байт указанного адреса.

ЗАПИСАТЬ БАЙТ СТАРШИЙ

ЗБС ппп

При выполнении этой инструкции содержимое старшего байта аккумулятора пересылается в старший байт указанного адреса.

ОБМЕНЯТЬ СОДЕРЖИМЫМ БАЙТ СТАРШИЙ И БАЙТ МЛАДШИЙ

ПББ ппп

При выполнении этой инструкции содержимое байта старшего пересылается в байт младший, а содержимое младшего байта в старший.

Пример.

Необходимо младший байт слова (010) переслать в младший байт слова (012), а старший байт содержимого слова (011) переслать в старший байт (012), в (013) слово записать содержимое (012) слова, только байт старший пересылается в младший, а младший байт в старший.

Информация, после выполнения программы:

Номер слова	Содержимое в восьмеричном формате
010	123 000
011	123 000

ПОБИТНОЕ УМНОЖЕНИЕ СЛОВ

ПУС ппп

Если условия в цепи истинны, то над каждой парой одноименных битов из аккумулятора и содержимого по адресу слова, производится логическая операция И.

Результат записывается в аккумулятор.

Пример.

```
НСТ 00
НБЛ 00
-II- I00 00
ЧТС 012
ПУС 013
ЗПС 014
НСТ 77
НПП 77
```

Номер слова	Содержимое в восьмеричном формате	
012	324 017	Исходная информация
013	135 302	Исходная информация
014	124 002	Результат выполнения инструкции ПУС

ПОБИТНОЕ СРАВНЕНИЕ СЛОВ

ПСС ппп

Если условия в цепи с инструкцией ПСС истинны, то над каждой парой бит из аккумулятора и содержимым по адресу слова производится логическая операция побитного сравнения, результат записывается в бит аккумулятора с тем же номером.

Пример.

```
НСТ 00
НБЛ 00
-II- I00 00
```


Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
1	НБЛ 00	
2	-II-II0 00	
3	-(S)-00000	
4	-()- 01314	
5	-II-00000	
6	-()-01315	
7	НСТ 77	
8	НПП 77	

РАСФИКСАЦИЯ ВЫХОДА

При замыкании контакта II000 (рис. 2.19) зажигается светодиод 01314, а светодиод 01315 не горит.

При размыкании контакта II0 00 светодиод 01314 гаснет.

При выключении питания, а затем при его восстановлении светодиоды 01314, 01315 не горят.

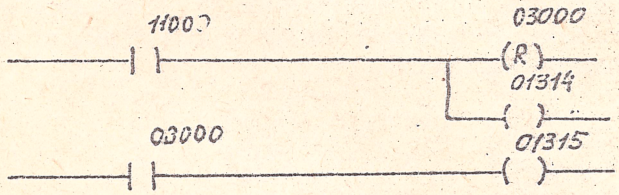


Рис. 2.19

Программа для цепей, изображенных на рис. 2.19:

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
1	НБЛ 00	
2	-II- II000	
3	-(R)- 03000	
4	-( )- 01314	
5	-II- 03000	
6	-( )-01315	
7	НСТ 77	
8	НПП 77	

ИНСТРУКЦИИ ВЕТВЛЕНИЯ

Группа инструкций ветвления включает три инструкции:

- начало одного ветвления;
- конец одного ветвления;
- перенос маркера в начало ветвления.

Эти инструкции служат для программирования параллельных логических ветвей в ПРП.

Часто выходное устройство должно включаться при более, чем одном сочетании входных условий. Инструкция ветвления применяется для формирования параллельных ветвей, формирующих более, чем одно условие включения выходной инструкции. Для включения выходной инструкции достаточно, чтобы логически замкнутый путь был установлен хотя бы в одной из параллельных ветвей.
Другими словами, между условиями параллельных ветвей производится логическая операция ИЛИ.

Если же ни в одной из ветвей не установлен замкнутый путь, то условия в цепи ложны и выход не включается.

На рис. 2.20 представлена цепь, имеющая разветвленное условие и выходную инструкцию.

В этой цепи выход включается при включении любого из входных битов: IOI OO или IOI OI. Как показано на рисунке логическое замыкание цепи может произойти по любому из двух путей, по любой из двух параллельных ветвей.

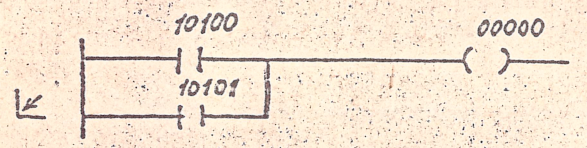


Рис. 2.20

На рисунке 2.20 изображена цепь реализующая логическую операцию ИЛИ: IOI OO ∨ IOI OI = 000 00.

Используя инструкции ветвления, можно программировать одну или несколько параллельных ветвей:

СЧЕТЧИК ЦИКЛОВ

СЧЦ ппп СЧЦ *nnn*

ПРК *kkkk* ПРС *mmmm*

Инструкция СЧЦ увеличивает в каждом скане программы содержимое слова, *nnn* , если результат равен I, и оставляет содержимое этого слова без изменения, если результат равен нолю.

Инструкция СБС сбрасывает содержимое счетчика в ноль.

Если значение счетчика достигло значения уставки, включается бит I6.

2.9. Инструкции для работы с данными

Все инструкции пословной обработки являются условными и выходными. Если условия в цепи истинны, то инструкции исполняются. Эти инструкции используются для пересылки данных из одного слова таблицы данных в другое слово, для копирования накопленных значений таймеров и счетчиков, для маскирования отдельных битов.

ЧИТАТЬ СЛОВО

ЧТС ппп
└──────────адрес слова

При выполнении инструкции ЧТС содержимое по адресу слова записывается в аккумулятор.

ЗАПИСАТЬ СЛОВО

ЗПС *nnn*

При выполнении инструкции ЗПС содержимое аккумулятора записывается по адресу слова.

Пример.

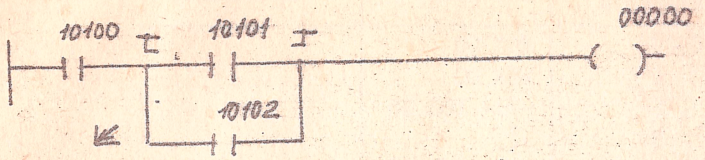
Исходная информация:

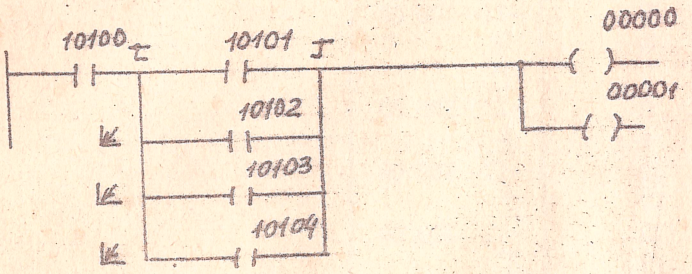
Номер слова	Содержимое в восьмеричном формате
OIO	I23 000
OII	000 000

Программа

```
НСТ 00
НБЛ 00
-II- I00 00
ЧТС OIO
ЗПС OII
НСТ 77
НПП 77
```

Номер инструкции	Мнемокод	Примечание
0003	СБС 004	
4	ПБЛ 01	
5	НБЛ 01	
6	-I/I- 001 I6	
7	ТВД 001	
8	ПРК 0010	
9	-II- 001 I6	
10	ПБЛ 02	
11	НБЛ 02	
12	-I/I- 003 I6	
13	ТВД 003	
14	ПРК 0020	
15	-II- 003 I6	
16	ПБЛ 01	
17	НСТ 01	
18	НБЛ 00	
19	-II- 001 I6	
20	СЧС 004	
21	ПРК 000 8	
22	-II- 004 I6	
23	-(R)- 004 I6	
24	ПБЛ 01	
25	НБЛ 01	
26	-II-003 I6	
27	С4В 004	
28	ПРК 0003	
29	-II-004 I6	
30	-(R)-004 I6	
31	ПБЛ 00	
32	НСТ 77	
33	НПП 77	





Можно программировать вложенные параллельные ветви, т.е. ветви, параллельные части неосновной ветви (рис. 2.21).

В ПК допускается обработка цепей, имеющих до восьми вложений. Если программа в ПК вводится с помощью МВ 96.02, то цепи должны содержать не более пяти вложений и последовательно включенных контактов до 8.

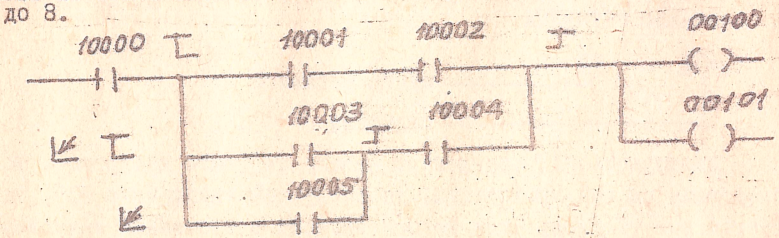


Рис. 2.21

Программа, реализующая цепь, изображенную на рис. 2.21:

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
I	НБЛ 00	
2	-II- I00 00	
3	⊢	
4	-II- I00 0I	
5	-II- I00 02	
6	⊬	
7	⊢	
8	-II- I00 03	
9	⊬	
I0	-II- I00 05	
II	⊢	
I2	-II- I00 04	
I3	⊢	
I4	-()- 00I 00	
I5	-()- 00I 0I	
I6	НСТ 77	
I7	КПП 77	

2.6. Инструкции структурирования

НАЧАЛО СЕГМЕНТА

НСТ пп

пп — номер сегмента

НСТ — мнемокод

Инструкция НСТ определяет начало сегмента программы. Все следующие за ней инструкции, до следующей инструкции НСТ, выполняются при условии, если данный сегмент в состоянии вычисления. Если сегмент в обнулении, то биты с операторами -( )- принимают значение логического ноля.

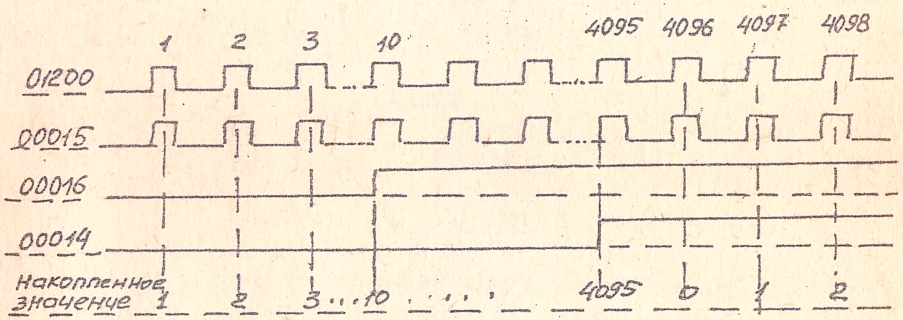
Информацию об условиях работы сегмента, процессор получает из области текущего состояния.

НАЧАЛО БЛОКА

НБЛ пп

пп — номер блока

Временная диаграмма



СЧЕТЧИК НА ВЫЧИТАНИЕ

СЧВ ппп СЧВ ппп

ПРС mmmm , ПРК кккк.

Когда цепь с инструкцией счета переходит из ложного в истинное состояние, накопленное значение уменьшается на единицу. Если счетчик начинает работу при условии, что накопленное значение больше уставки, то I6 бит служебных разрядов не включается, а когда накопленное значение становится равным уставки, то I6 бит включается. Если счетчик начинает работу при условии, что накопленное значение меньше уставки, то вкл. I6 бита после перехода через нулевое значение, накопленное значение ставится равным 4095 и включаются биты I4 и I6.

Бит I7 служит битом разрешения счета на вычитание. Он включается, когда истинны условия цепи.

Для организации реверсивного счетчика необходимо счетчик на сложение и счетчик на вычитание запускать на одном слове и они должны работать в разных блоках.

Пример организации реверсивного счетчика:

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
1	НБЛ 00	
2	ОСТ 0I	

Счетчик на сложение используется для счета переходов цепи из ложного в истинное состояние. Эти переходы могут быть вызваны проходом деталей перед детектором или включением конечного выключателя. Отсчет сохраняется при переходе цепи в ложное состояние, сброс счетчика происходит после выполнения инструкции СБС, имеющей тот же адрес, что и счетчик.

Когда цепь с инструкцией счета переходит из ложного в истинное состояние, накопленное значение увеличивается на единицу. Когда накопленное значение достигнет уставки, возбуждается I6 бит слова, на котором запущен счетчик на сложение.

Бит I5 служит битом разрешения счета на сложение. Он включается, когда истинны условия цепи.

Инструкция счета на сложение предназначена для счета до своей уставки. Если счет продолжается, после достижения уставки, а накопленное значение переходит к свое предельное значение 4095, то включается бит I4. Биты I4 и I6 сбрасываются инструкцией СБС с тем же адресом, что и инструкция счетчика на сложение.

Пример программы с использованием счетчика на сложение.

Исходные данные: в слово 0020 записать константу 0010_{10}.

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
1	НБЛ 00	
2	ОСТ 01	
3	СБС 000	
4	ПБЛ 01	
5	НБЛ 01	
6	-I/I- 010 I6	
7	ТВД 010	
8	ПРК 0001	
9	-II- 010 I6	
10	-()- 012 00	
11	НСТ 01	
12	НБЛ 00	
13	-II- 010 I6	
14	СЧС 000	
15	ПРС 0020	
16	НСТ 77	
17	НПП 77	

Инструкция определяет начало блока в данном сегменте. Все следующие за ней инструкции (за исключением инструкций НСТ, НБЛ выполняются при условии, если данный блок активный. Информацию о том, какой блок активный в данном сегменте процессор получает из области текущего состояния. Инструкция НБЛ устанавливает результат в единицу. Это значит, что если за инструкцией НБЛ следует инструкция, включить выход, то данный выход возбудится.

Последовательность блоков в сегменте может быть произвольной.

ПЕРЕЙТИ НА БЛОК

ПБЛ ПП ——— номер блока

Когда цепь, содержащая инструкцию ПБЛ, становится истинной, процессор записывает в область текущего состояния, в байт соответствующий обрабатываемому сегменту, номер активного блока, указанный в операндной части инструкции ПБЛ. Следующие за ПБЛ инструкции не выполняются.

Когда цепь, предшествующая ПБЛ, ложна, процессор не выполняет инструкцию ПБЛ, а выполняет следующие за ней инструкции.

На рис. 2.22 приведен пример, поясняющий принцип сканирования.

ЗАКРЫТЬ СЕГМЕНТ

ЗСТ ПП ——— номер сегмента

Когда цепь, предшествующая ЗСТ, истинна, процессор устанавливает в состояние игнорирования тот сегмент, который указан в операндной части инструкции (за исключением нулевого сегмента).

Если ЗСТ имеет в операндной части номер сегмента тот же, что и номер сегмента в котором находится данная инструкция, то на протяжении одного скана, в котором инструкция ЗСТ выполнялась, инструкции следующие за ней так же будут исполняться. В следующем скане, процессор не выполнит ни одной инструкции данного сегмента и так будет продолжаться до тех пор, пока не выполнится инструкция ОСТ с тем же номером. Вычисления в сегменте начнутся в блоке, который работал в момент выполнения инструкции ЗСТ.

ОТКРЫТЬ СЕГМЕНТ

ОСТ ПП ——— номер сегмента

Когда цепь, содержащая инструкцию ОСТ, становится истинной, то сегмент, указанный в операндной части, устанавливается в состояние вычисления. В байте области текущего состояния данного

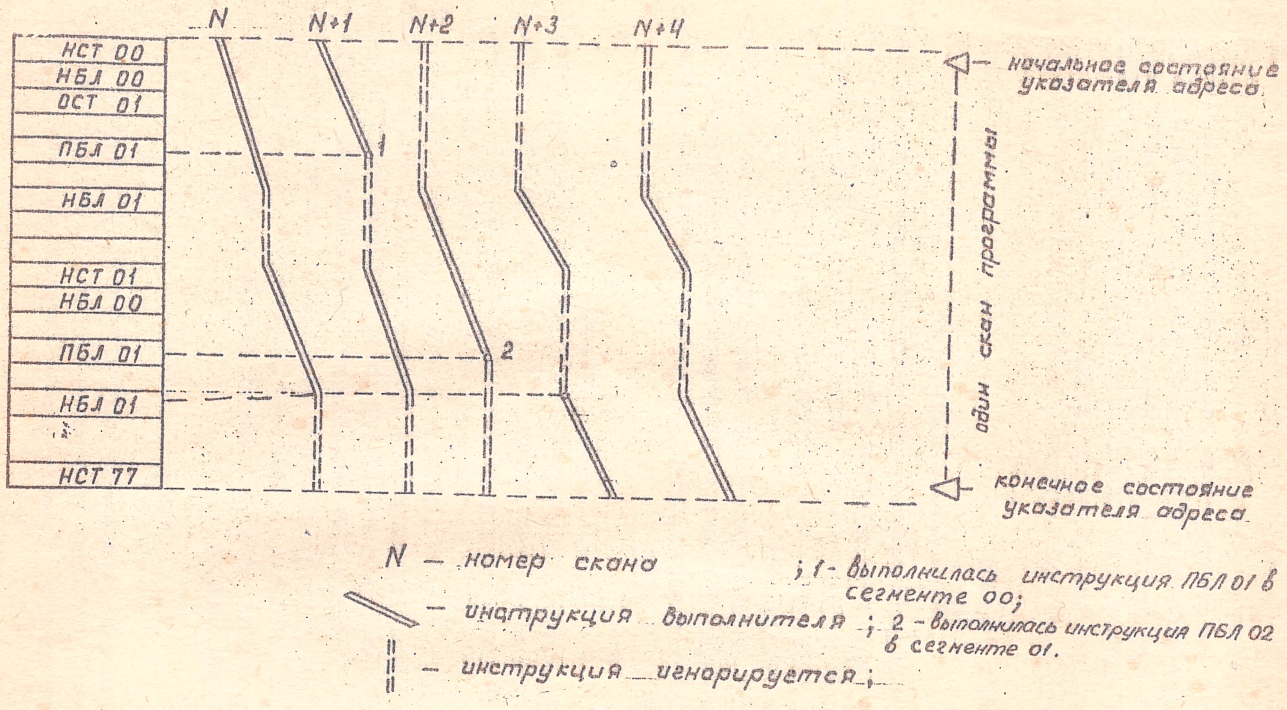


Рис. 2.22

Номер инструкции	Мнемокод	Примечание
0008	ПРК 0010	
0009	-II- 001 I6	
0010	ПБЛ 02	
0011	НБЛ 02	
0012	-I/I- 003 I6	
0013	ТВД 003	
0014	ПРК 0020	
0015	-II- 003 I6	
0016	ПБЛ 01	
0018	НСТ 01	
0018	НБЛ 00	
0019	-II- 003 I6	
0020	ТНД 000	
0021	ПРК 0300	
0022	-II- 000 I6	
0023	ПБЛ 01	
0024	НБЛ 01	
0025	-I/I- 002 I6	
0026	ТНД 002	
0027	ПРК 0040	
0028	-II- 002 I6	
0029	ПБЛ 00	
0030	НСТ 77	
0031	НПП 77	

СЧЕТЧИК НА СЛОЖЕНИЕ
СЧС ппп
ПРС m m m ,
СЧС n n n
ПРК к к к к .

Если в период работы накапливающего таймера условия в цепи становятся ложными, то это не приводит к потере накопленного значения. Напротив, когда цепь снова станет истинной, таймер продолжит счет времени с той величины, которая была накоплена к моменту остановки. Благодаря способности сохранять накопленное значение этот таймер может измерять суммарное время истинного состояния его цепи.

Накопленное значение накапливающего таймера может быть сброшено инструкцией сброса (СБС) с тем же адресом, что и таймер.

Бит I6 включается, когда накопленное значение достигает уставки. Этот бит может быть сброшен только инструкцией сброса.

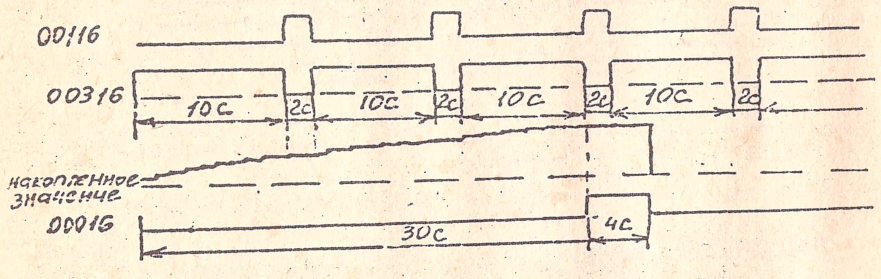
Бит I7 включается, когда условия в цепи становятся истинными и выключается, когда эти условия становятся ложными.

Бит I5 отображает дискрету времени.

Пример.

Через 30 с суммарного времени истинного состояния цепи, выдать сигнал длительностью 4с.

Временная диаграмма



Программа

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
0001	НБЛ 00	
0002	ОСТ 01	
0003	СБС 000	
0004	НБЛ 01	
0005	НБЛ 01	
0006	-I/I- 001 I6	
0007	ТВС 001	

сегмента бит 5 шесть и бит семь принимают истинное значение.

СБРОСИТЬ СЕГМЕНТ

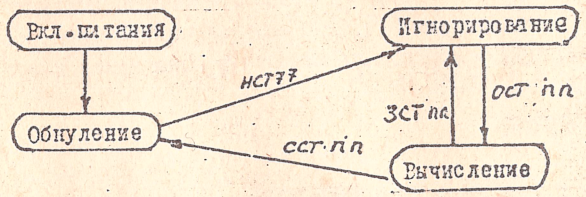
ССТ пп — номер сегмента

Когда цепь, содержащая инструкцию ССТ, становится истинной, то сегмент, указанный в операндной части, устанавливается в состояние обнуления, т.е. процессор исполняет инструкции независимо от вычислений, с результатом равным нулю. Байт текущего состояния становится равным нулю.

После выполнения команды НСТ 77, сегмент устанавливается в состояние игнорирования. Инструкцию ССТ необходимо писать в программе до начала соответствующего сегмента, иначе обнуления не произойдет.

Из состояния игнорирования, сегмент выводится командой ОСТ и вычисление начинается с нулевого блока.

Взаимосвязь команд структурирования:



пп - номер сегмента

УСТАНОВИТЬ НОМЕР СЕКТОРА

УНС пп — номер сектора

Отказы модулей фиксируются в словах 3716(7716 - для МС59.06) и 3717(7717 - для МС59.07). Чтобы получить доступ в слово, например, 3716, необходимо записать:

```
    ⋮
УНС 07
ЧТС 316
УНС 00
ЗПС W        W - слово, в котором
    ⋮          будут анализироваться
             биты отказов.
```

Для доступа в слово 7716, необходимо записать:

УИС 17
ЧТС 316
УИС 00
ЗЛС W
⋮

2.7. Инструкции для работы с подпрограммами

Одинаковые фрагменты в программе можно организовать, как одну подпрограмму и вызывать в нужные моменты. Обращения к подпрограммам (вызовы) находятся в теле рабочей программы, а подпрограммы располагаются после конца программы (после инструкции НСТ 77). В подпрограмме не допускается применять инструкции структурирования.

ВЫЗВАТЬ ПОДПРОГРАММУ

ВПП пп — номер подпрограммы

Когда цепь, содержащая инструкцию ВПП становится истинной, процессор вычисляет подпрограмму с соответствующим номером, после конца подпрограммы, процессор возвращается на выполнение инструкции следующей за инструкцией ВПП.

НАЧАЛО ПОДПРОГРАММЫ

НПП пп — номер подпрограммы

Инструкции ВПП соответствует инструкция НПП с тем же номером. После НПП следует подпрограмма.

КОНЕЦ ПОДПРОГРАММЫ

НБЛ 77

Подпрограмма обязательно должна заканчиваться инструкцией НБЛ 77.

КОНЕЦ МАССИВА ПОДПРОГРАММ

НПП 77

Рабочая программа завершается инструкцией НПП 77.

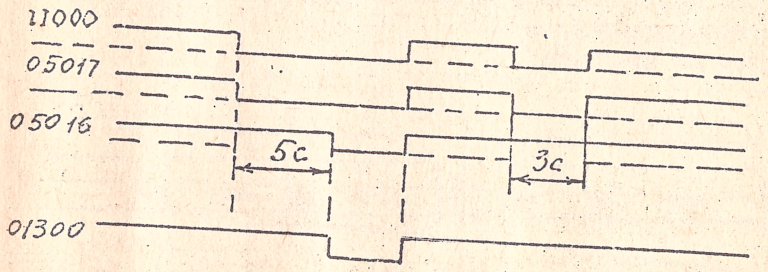
Пример программы с двумя подпрограммами:

```
НСТ 00
НБЛ 00
-II- II2000
-()- 006 04
-II- II2 0I
```


Программа

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
I	НБЛ 00	
2	-II- II0 00	
3	ТОД 050	
4	ПРК 0050	
5	-II- 050 I6	
6	-()- 0I3 C0	
7	НСТ 77	
8	НПП 77	

Временная диаграмма



НАКАПЛИВАЮЩИЙ ТАЙМЕР

ТНД – таймер накапливающий с дискретностью отсчета времени 0,1с,

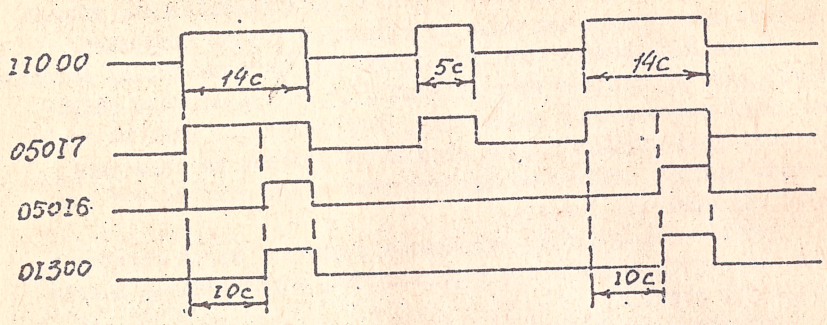
ТНС – с дискретностью отсчета времени 1с.

ТНД nnn
ПРК kkkk ,
ТНД nnn
ПРС mmmm ,

ТНС nnn
ПРК kkkk ,
ТНС nnn
ПРС mmmm .

Накапливающий таймер считает дискреты времени, когда его управляющая цепь логически истинна, и сохраняет свое накопленное значение, если условия в цепи становятся ложными, а также при потере питания.

Временная диаграмма:



ТАЙМЕР С ЗАДЕРЖКОЙ НА ОТКЛЮЧЕНИЕ

ТОД – таймер с задержкой на отключение с дискретностью отсчета времени 0,1 с;

ТОС – с дискретностью отсчета 1с.

ТОС nnn	ТОС nnn
ПРК kkkk ,	ПРС mmmm ,
ТОД nnn	ТОД nnn
ПРК kkkk ,	ПРС mmmm .

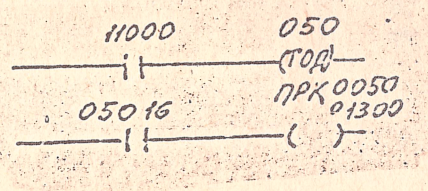
Таймер с задержкой на отключение считает количество временных дискрет и хранит их счет в накопленном значении.

Таймер с задержкой на отключение начинает отсчет, когда условия в его управляющей цепи становятся ложными и продолжает счет пока накопленное значение не достигнет уставки.

Бит I7 отслеживает ложное и истинное состояние управляющей цепи. Бит I5 отображает дискрету времени. Бит I6 – завершение операции.

Пример с использованием таймера на отключение:

выход 013 00 должен отключиться через 5с, после размыкания контакта 110 00.




```
НПП 01
-II- II2 02
-()- 006 00
-II- II2 03
НПП 02
-II- II2 04
-()- 006 01
НСТ 77
НПП 01
-II- II2 05
-II- II2 06
-II- II2 07
-()- 006 02
НБЛ 77
НПП 02
-II- II2 I0
-()- 006 03
НБЛ 77
НПП 77
```

При включении бита II2 01 произойдет вызов подпрограммы I. Процессор решит функцию: .II205 .II206 + .II207 = 00602; и по команде НБЛ 77 вернется на чтение контакта II2 02.

2.8. Инструкции таймеров и счетчиков

Группа инструкций таймеров и счетчиков включает следующие инструкции:

- таймер с задержкой на включение;
- таймер с задержкой на отключение;
- накапливающий таймер;
- счетчик на вычитание;
- счетчик на сложение;
- счетчик циклов.

Благодаря этим инструкциям ПК имеет возможности аналогичные реле времени, полупроводниковым, временным и счетным устройствам. В ПК таймеры и счетчики реализованы программно с помощью *резидентного* программного обеспечения. Тем не менее они должны программироваться пользователем, который задает условия их работы.

Таймеры отсчитывают интервалы времени, счетчики считают события по алгоритмам, определяемым логикой программы пользователя.

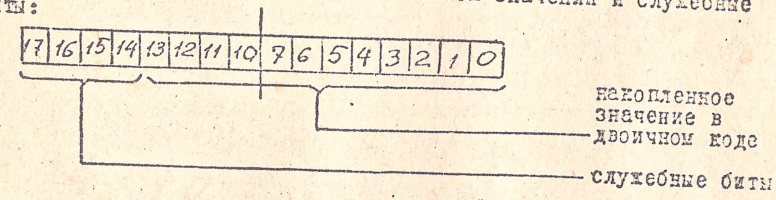
Каждая инструкция таймера и счетчика связана с уставкой – заранее заданным числом. Когда накопленное значение достигает уставки, включается бит состояния, который может быть использован для управления выходным устройством.

Накопленное значение – текущее количество базовых интервалов времени, которое сосчитал таймер или количество событий, сосчитанных счетчиком.

Уставка и накопленное значение для таймеров и счетчиков могут изменяться от 0 до 4095.

Уставка таймера и счетчика может задаваться в виде константы, которая в программе следует за инструкцией таймера – ПРК, либо адресом слова, где находится уставка – ПРС.

По адресу, который следует за инструкцией таймера или счетчика, хранится информация о накопленном значении и служебные биты:



Дискретность счета – это базовые интервалы времени.

Для ПК доступны следующие значения базового интервала: Ic; 0,Ic

Дискретность счета таймеров, задается в их мнемонике, например, таймер с задержкой на включение и с дискретностью счета Ic – ТВС, а ТВД этот же таймер с дискретностью счета 0,Ic.

Если в программе используется таймер с дискретностью счета времени 0,I с, то время цикла не должно превышать 0,05 с. Для таймера с дискретностью счета I с время цикла не должно превышать 0,5 с. *Таймеры считают только в открытом сегменте и активном блоке.*

ТАЙМЕР С ЗАДЕРЖКОЙ НА ВКЛЮЧЕНИЕ

ТВД – дискретность отсчета времени 0,Ic,

ТВС – дискретность отсчета времени Ic.

ТВД *nnn* ; ТВД *nnn* ; ТВС *nnn* ; ТВС *nnn*

ПРК *kkkk* ; ПРС *mmmm* , ПРК *kkkk* ; ПРС *mmmm* ,

где *nnn* – слово, на котором закодирован таймер;

kkkk – константа в десятичном формате от 0 до 4095;

mmmm – слово, в которое записана константа.

Когда условия в управляющей цепи таймера становятся истинными, таймер начинает считать с указанной дискретностью, наращивая свое накопленное значение на единицу в каждом интервале.

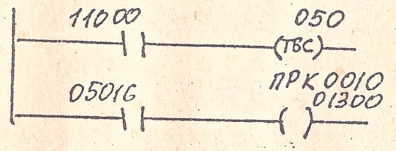
Когда накопленное значение достигает уставки, таймер останавливает счет и включает шестнадцатый (I6) бит завершения операции. Этот бит остается включенным, пока условия в цепи не станут ложными. Если условия в цепи становятся ложными до окончания отсчета времени, таймер останавливается и его накопленное значение сбрасывается в ноль.

Бит семнадцатый (I7) включается в момент, когда условия в цепи становятся истинным и сбрасывается, когда условия в цепи становятся ложными.

Бит пятнадцатый (I5) – отображает дискрету времени.

Пример с использованием таймера на включение:

выход 013 00 должен включиться через 10с, после замыкания контакта 110 00



Программа:

Номер инструкции	Мнемокод	Примечание
0000	НСТ 00	
I	НБЛ 00	
2	-II- 110 00	
3	ТВС 050	
4	ПРК 0010	
5	-II- 050 I6	
6	-( )- 013 00	
7	НСТ 77	
8	НПП 77	